AN AB INITIO LNCRNA IDENTIFICATION AND FUNCTIONAL ANNOTATION TOOL BASED ON DEEP LEARNING

A Dissertation
Presented to
The Academic Faculty

By

Cheng Yang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Department of Biomedical Engineering

Georgia Institute of Technology, Emory University and Peking University

May 2018

AN AB INITIO LNCRNA IDENTIFICATION AND FUNCTIONAL ANNOTATION TOOL BASED ON DEEP LEARNING

Approved by:

Dr. Huaiqiu Zhu, Advisor
Department of Biomedical Engineering
Peking University

Dr. May D. Wang, Co-advisor
Department of Biomedical Engineering
Georgia Institute of Technology and Emory University

Dr. Zuhong Lu
Department of Biomedical Engineering
Peking University

Dr. Jianzhong Xi
Department of Biomedical Engineering
Peking University

Dr. Minghua Deng
School of Mathematical Sciences
Peking University

Date Approved: December 7, 2017

To my family.

ACKNOWLEDGEMENTS

This part is intended to thank all who have helped and supported me during my research. Without them, this dissertation could not be done.

I would like to extend my gratitude to my advisor Dr. Huaiqiu Zhu, for his invaluable tutorship, vision, and rigorous training. It is Dr. Zhu who led me to the field of bioinformatics. I used to experience a very depressed period, however, Dr. Zhu's encouragements strengthened me to persevere and accomplish my research.

I had a wonderful experience at Georgia Tech and I would like to extend my gratitude to my co-advisor Dr. May D. Wang at GT, for her valuable guidance, kindly encouragements and rigorous instructions. I learned so much from Dr. Wang, such as efficient communication, hardworking, and the most impressive is to persue the perfection.

I would like to thank my collegues and friends, who have helped me in various ways. They are Yongchu Liu, Wenqi Wu, Binbin Lai, Feifei He, Luying Liu, Longshu Yang, Qi Wang, Yaoguang Xing, Xiaoqi Wang, Peng Zhai, Zhe Wang, Qiuyue Wang, Xiao Guo, Li Qu, Fangmin Tian, Xiqoqing Jiang, Luotong Wang, Congmin Xu, Peihong Wang, Mo Li, Zhongjie Xie, and Man Zhou at PKU lab; John Phan, Chanchala Kaddi, Janani Venugopalan, Leo Wu, James Cheng, Zhimin Han, Ryan Hoffman, Li Tong, Ying Sha, and Hang Wu at GT lab. They are all very smart and I learned much from them through discussions and collaborations. Gratitude is also extended to Bo Zhao, Hao Luo, Kedi Zhou, Hao Xie, Xiyu Li, and Jinyang Wang for for their help during my study at GT.

Finally, I would like to thank my parents and my wife Haoling Xie, for their unconditional support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

As an important component of transcriptome, long noncoding RNA (lncRNA) refers to noncoding RNA whose length is above 200 nt. lncRNAs play important biological roles, such as dosage compensation, genomic imprinting, cell differentiation and have been implicated in human disease. Although some lncRNAs have been characterized, the functions of most lncRNAs currently remain unclear. To characterize lncRNAs, identifying lncRNAs and inferring their functions is necessary. In this dissertation, we propose a novel lncRNA identification and functional annotation tool named LncADeep.

lncRNA identification refers to distinguishing lncRNAs from mRNAs. As a prevalent method for studying lncRNAs, current RNA-seq techniques tend to generate short reads, impeding accurate full-length transcript assembly. Thus, lncRNA identification is further complicated by partial-length mRNAs reconstructed from RNA-seq short reads, since partial-length mRNAs truncated at 5' and/or 3' end can lead to incomplete coding sequences (CDS), which are prone to be misclassified as lncRNAs. However, most tools focus on identifying lncRNAs from full-length transcripts, and it is necessary to develop a method for lncRNA identification from transcripts including full- and partial-length. Herein, we propose LncADeep, which identifies lncRNAs by integrating sequence intrinsic and homology features based on deep belief networks. In particular, LncADeep constructs two models for lncRNA identification, one targeting full-length transcripts and the other targeting transcripts including full- and partial-length. Benchmarked with eleven state-of-the-art lncRNA identification tools (i.e., CPC, CPC2, CPAT, CNCI, COME, lncRScan-SVM, lncRNA-MFDL, longdist, lncScore, FEElnc and PLEK) with 10-fold cross-validation, our tool LncADeep has outperformed them on full-length human transcripts with an accuracy of 97.7% (sensitivity 98.1%, specificity 97.2%), and transcripts including full- and partial-length with an accuracy of 94.2% (sensitivity 93.8%, specificity 94.5%). Besides, LncADeep still outperformed the above tools on cross-species lncRNA identification on

full-length mouse transcripts with an accuracy of 96.7% (sensitivity 97.0%, specificity 96.3%) and transcripts including full- and partial-length with an accuracy of 94.2% (sensitivity 95.1%, specificity 93.3%). Results have shown that LncADeep is a robust lncRNA identification tool, which outperforms state-of-the-art tools on full-length transcripts, transcripts including full- and partial-length, and cross-species lncRNA identification.

Since lncRNA-protein interactions play the crucial roles in the functioning of lncRNAs, we propose to infer the functions through predicting lncRNA-protein interactions. Herein, LncADeep predicts lncRNA-protein interactions using sequence and structure features based on deep neural networks, where LncADeep has achieved a higher accuracy of 90.8% (sensitivity 97.0%, specificity 85.4%) than state-of-the-art tools including lncPro, RPISeq, RPI-pred, rpiCool, and IPMiner. To infer the functions of lncRNAs, LncADeep conducts KEGG and Reactome pathway enrichment analysis and functional module detection with the predicted interacting proteins of lncRNAs. Compared with the above five tools, not only can LncADeep predict lncRNA-protein interaction with better performance, but also provide functional annotations for lncRNAs automatically. We then annotate the 27,384 lncRNAs collected by GENCODE database, and LncADeep annotated each lncRNA with with an average of 25 KEGG and 67 Reactome pathways. In contrast, using IPMiner's predicted proteins for enrichment analysis, we only obtained an average of 5 KEGG and 28 Reactome pathways for each lncRNA, which is less than LncADeep's annotation. Moreover, case studies show that LncADeep's functional annotations for lncRNAs comply with their known functions.

In summary, we develop a novel lncRNA identification and functional annotational tool, LncADeep (http://cqb.pku.edu.cn/ZhuLab/LncADeep), based on deep learning algorithms. First, LncADeep outperformed state-of-the-art lncRNA identification tools on full-length transcripts and transcripts including full- and partial-length. Since transcripts reconstructed from RNA-seq dataset are composed of full- and partial-length, LncADeep is particularly useful for RNA-seq community. Second, LncADeep outperformed state-of-the-art tools

on predicting lncRNA-protein interactions. Based on the predicted lncRNA-protein interactions, LncADeep provides rich functional annotations, conforming with the known functions, for lncRNAs. To our knowledge, LncADeep is the first tool which can identify lncRNA and annotate lncRNAs functions automatically. Currently, there are still a large amount of lncRNAs to be identified, while the functions of most lncRNAs remain unclear. We expect that not only can LncADeep contribute to identifying lncRNAs, but also provide helpful functional information for investigating the associations among lncRNAs, gene regulation and diseases, and then facilitate the large-scale automatic genome annotation.

CHAPTER 1
INTRODUCTION

This chapter intends to introduce the motivation and background for this dissertation, which is mainly about LncADeep: an *ab initio* long noncoding RNA (lncRNA) identification and functional annotation tool based on deep learning methods. lncRNA is a critical component of transcriptome, playing important biological roles and attracts much attention. As an approach to studying transcriptome and with the development of high-throughput sequencing technologies, RNA sequencing (RNA-seq) has clear advantages over some other approaches such as microarray, and has become the preferred method. To characterize lncRNAs from RNA-seq data, lncRNA identification and functional annotation is necessary. Herein, we first give a introduction to RNA-seq, and then introduce the background of lncRNA. As we use deep learning methods for lncRNA identification and functional annotation, we then give a brief description for neural networks and deep learning methods.

1.1 RNA-seq for studying transcriptomics

The term transcriptome was originally proposed by Charles Auffray in 1996, and defined as the entire sets of transcripts [1]. As a intermediate of the flow of the genetic information, transcriptome consists of various categories of transcripts, such as protein-coding RNAs (or mRNAs), noncoding RNAs and small RNAs. One key aim of transcriptomics is to catalogue these categories of transcripts and investigate their transcriptional structures, such as alternative splicing patterns, 5' and 3' ends [2, 3]. In addition, another important aim of transcriptomics is to quantify the transcript expression under various conditions, which can help to, for example, reveal the differentially expressed transcripts among tissues, between sexes, etc [4, 3]. Characterizing transcriptome can not only help to "interpret the functional elements of the genome and reveal the molecular constituents of cells and tissues", but also

"understand development and disease" [3].

1.1.1 Technologies for studying transcriptomics

To characterize transcriptome, several technologies have been developed. The early technologies are low-throughput, such as Serial/Cap Analysis of Gene Expression (SAGE/CAGE) [5, 6], which are based on the expressed sequence tags (ESTs) using low-throughput Sanger sequencing technologies. However, these low-throughput technologies have largely been overtaken by high-throughput technologies like microarray and RNA sequencing (RNA-seq), which are dominant technologies for characterizing transcriptome and can provide more helpful information.

Microarray is composed of fixed probes (such as short oligonucleotides or cDNA) arrayed on a solid substrate (such as glass slide), to which the fluorescently labelled transcripts can bind and then generate fluorescence signals [7, 8]. After hybridisation and washes, the abundance of transcripts can be determined by the total strength of fluorescence signals at each probe location. However, to synthesize the probes, microarray requires prior knowledge, such as a reference genome or transcriptome, limiting its application on new organisms whose reference genome or transcriptome is not available.

Unlike microarray which highly depends on prior knowledge, RNA-seq can directly determine the sequence using high-throughput sequencing technologies, such as Illumina [9]. A typical RNA-seq experiment [10] is illustrated in Figure 1.1. First, after being isolated from tissues, transcripts are converted to RNA fragments. Second, RNA fragments are reverse-transcribed to complementary DNAs (cDNAs). Third, cDNAs are used for library preparation, including ligating adapters and PCR amplification. Finally, with high-throughput sequencing, the resulting short sequence reads can be used for quantifying expression level with data analysis.

Compared with microarray, RNA-seq has several advantages. (i) Resolution: The resolution of RNA-seq is single base, which is particular helpful for discovering single nu-

cleotide variants, while that of microarray ranges from several to 100 nt [11]. (ii) Background noise: The observed intensity of each probe in microarray is mixed up with specific and nonspecific binding, and optical noise, which requires complicated methods for background noise adjustment [12], while RNA-seq does not suffer this problem. (iii) Detecting highly and lowly expressed genes: The dynamic range of detecting gene expression level of RNA-seq is almost unlimited, while that of microarray is limited to a few hundred [13]. Therefore, because of these clear advantages over microarray, RNA-seq has become a prevalent method for transcriptomics.

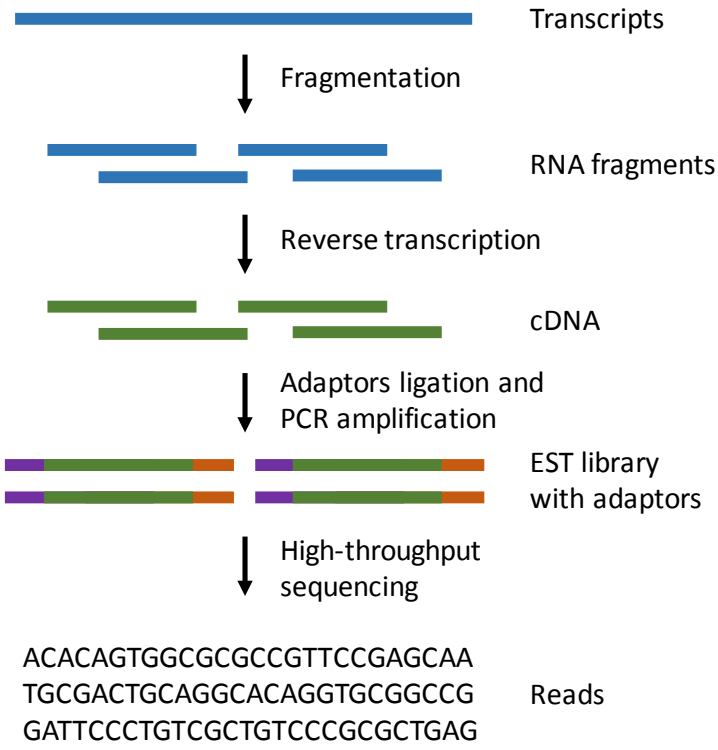


Figure 1.1: The flowchart of a typical RNA-seq experiment.

1.1.2 RNA-seq data analysis

To mining meaningful information from RNA-seq data, the data analysis is necessary. A typical RNA-seq data analysis pipeline includes four steps, (i) sequence alignment and assembly, (ii) expression quantification, (iii) expression normalization, and (iv) differential expression analysis.

Sequence alignment refers to aligning the short reads to a reference genome or transcriptome, which is usually available for model organisms. Aligning short reads to the reference genome can help to discover novel transcripts which have not been annotated. However, because of the splice junctions of exons in eukaryotic genome, it is necessary to use spliced aligners. There are many spliced aligners, such as Tophat2 [14], STAR [15], MapSplice [16], HISAT [17], GEM [18] and PASSION [19], which can detect novel splice junctions. If novel transcripts are not interested, aligning reads to a reference transcriptome using un-spliced aligners will suffice. There are also several un-spliced aligners, such as Bowtie2 [20], BWA [21], WHAM [22]. Regardless of spliced or un-spliced aligners, because of the short length, reads can be aligned to multiple locations of the genome (or transcriptome). The ambiguity resulting from multiple-aligned reads can affect the subsequent steps, such as assembly and expression quantification [23].

For RNA-seq, sequence assembly refers to reconstructing transcript through aligning and merging the short reads. Sequence assembly methods can be divided into two categories, genome-guided assembly and *de novo* assembly [24]. Genome-guided assembly requires sequence alignment and genome annotation to reconstruct transcripts, which has been a prevalent approach for model organisms, and several pipelines such as Tophat/Cufflinks [25] and HISAT/StringTie [26] are in common use. As previously mentioned, genome-guided assembly can be biased by multiple-aligned reads [23]. *De novo* assembly does not require prior knowledge, which is especially helpful for non-model organisms without reference genome. Several *de novo* assembly methods such as Trans-ABySS [27] and Trinity [28] are available, and both of them are based on constructing de Bruijn graphs with the *kmers* of short reads [27, 28], which can reconstruct a transcript without reference genome. However, *de novo* assembly methods tend to be time- and resource-consuming [28]. Although sequence assembly methods provide approaches for reconstruct transcripts from RNA-seq dataset, reconstructing full-length transcripts still remains a challenge. According to an assessment of various RNA-seq assembly methods,

the best-performing unguided assembly tools can identify at most 59% full-length mRNAs from *C. elegans* and only 21% from *H. sapiens* datasets, and over 30% in *C. elegans* and 60% in *H. sapiens* reconstructed transcripts are of partial-length [29], where partial-length transcripts can lead to partial-length coding sequence and thus affect the classification of transcripts into mRNAs or noncoding RNAs.

One of the key application of RNA-seq is gene (or transcript) expression estimation. After aligning reads to a reference genome or transcriptome, the next step is to quantify gene expression. There are also many tools for expression quantification such as HTseq [30], RSEM [31], Cufflinks [25], and StringTie [32]. HTseq can count the reads mapping to genes or transcripts and return raw read counts. However, the raw read counts can not be used alone for quantifying expression level among samples, since these values are influenced by many factors like transcript length and sequencing depth and biases [33]. Therefore, to generate comparable expression quantification, expression normalization is necessary [34]. For example, Reads Per Kilobase Million (RPKM) [2] and Transcripts Per Kilobase Million (TPM) are two common normalization methods [34, 35], which can be computed as Equations 1.1 and 1.2, respectively. Unlike HTseq, which outputs raw read counts, RSEM, Cufflinks, and StringTie use sophisticated algorithms to estimate expression levels. For example, RSEM computes maximum likelihood abundance estimates for transcripts using the Expectation-Maximization (EM) algorithm, which can address the uncertainty introduced by multiple-aligned reads [31], while the time cost of RSEM can be very expensive. If the only target is gene expression estimation, sequence alignment is not necessary, and thus several alignment-free (or light alignment) quantification methods have been developed based on bloom filter, such as Sailfish [36], RNA-skim [37], Salmon [38] and kallisto [39], which are much faster than the above methods.

$$\begin{aligned} \text{Scaling factor} &= \frac{\text{Number of total reads}}{1,000,000} \\ \text{RPKM} &= \frac{\text{Read counts of gene}}{\text{Scaling factor}} \times \frac{1}{\text{Length of gene}} \end{aligned} \tag{1.1}$$

$$\begin{aligned} \text{RPK} &= \frac{\text{Read counts of gene}}{\text{Length of gene}} \\ \text{Scaling factor} &= \frac{\sum_{\text{Sample}} \text{RPK}}{1,000,000} \\ \text{TPM} &= \frac{\text{RPK}}{\text{Scaling factor}} \end{aligned} \tag{1.2}$$

After gene expression quantification and normalization, the next step is to identify differentially expressed genes (DEGs) among samples. There are also many DEG detection tools, such as DEGseq [40], DEseq [41], edgeR [42], and Ballgown [43]. Although the aforementioned normalization methods (RPKM and TPM) can help to normalize the difference of sequence depth and library size between samples, it might be not adequate for DEG detection [44]. For example, even if the sequence depth and library size are identical between samples, highly expressed genes in a sample can repress the counts for other genes, when compared to a sample where genes are more evenly distributed, many genes can be falsely identified as differentially expressed [44]. Therefore, DEG detection tools are usually shipped with further normalization methods, such as edgeR (released with TMM) [45] and DEseq. Most of DEG detection tools assume that the gene expression conforms Poisson or Negative Binomial distribution, and then fit and test for differential expression. *Soneson, C.* benchmarked eleven DEG tools, however, no optimal method was found under all circumstances, and thus the choice of method depends on various conditions [44].

1.2 Introduction to long noncoding RNA

Facilitated by high-throughput sequencing technologies, the investigations on genomics and transcriptomics have shed light on the previously known "junk DNA", which accounts for a large proportion of the genome [46] and used to represent "any DNA sequence that does not play a functional role in development, physiology, or some other organism-level capacity" [46]. However, it has been estimated that approximately three-quarters of human genome is pervasively transcribed, of which less than 2% encodes for proteins [47, 48, 49]. In other words, a large fraction of the genome is transcribed into noncoding RNAs, and the majority are lncRNAs. Studies have revealed that lncRNAs play important biological roles in transcription, dosage compensation, genomic imprinting, cell differentiation [47, 50], and have been implicated in various human disease such as cancers (e.g., breast cancer, melanoma) [51], indicating the junk DNA is not "junk". As a key component of transcriptome, long noncoding RNA (lncRNA) is defined as noncoding RNA whose length is above 200 nt, and some "classic" ncRNAs are not included, such as ribosomal RNA (rRNA) and transfer RNA (tRNA) [52].

1.2.1 The functions of lncRNA

The functions of lncRNA are diverse and complicated. Initially, lncRNAs were considered to be primarily associated with the epigenetic regulation [53], such as well-studied HOTAIR and XIST. For example, interacting with Polycomb Repressive Complex 2 (PRC2) and the CoREST-LSD1 complex, lncRNA HOTAIR specifies the pattern of histone modification on target genes and is required for PRC2 occupancy and histone H3 lysine-27 trimethylation of HOXD locus [54, 55]. As an essential lncRNA required for X chromosome inactivation, lncRNA XIST can interact with 81 proteins from chromatin modification, nuclear matrix, and RNA remodeling pathways to induce gene silencing [56]. Recent studies revealed that the functions of lncRNAs are more than epigenetic regulation. Con-

sidering the processes that many lncRNAs experience during their lifetime, the functions of lncRNAs can be involved in transcriptional regulation (e.g., DEANR1), post-transcriptional processing (e.g., lincRNA-p21), and some non-regulatory functions (e.g., PANDA) [53]. lncRNA DEANR1 can positively regulate the expression of the endoderm factor FOXA2 and then facilitate endoderm differentiation [57]. lncRNA lincRNA-p21 can selectively associated with JUNB and CTNNB1 mRNAs and represses their translation, acting as a post-transcriptional inhibitor of translation [58]. Interacted with the transcription factor NF-YA, lncRNA PANDA can impede the activation of pro-apoptotic gene expression [59].

lncRNAs can interact with DNAs, RNAs, and proteins, among which the lncRNA-protein interactions are confirmed to be the first and principle interactions and functions for lncRNAs, such HOTAIR and XIST [60, 61]. Several mechanisms of lncRNA binding to proteins have been discussed, where lncRNAs act as decoys, scaffolds, and guides [50, 62]. Interacting with DNA-binding proteins, lncRNAs can prevent the access of proteins to DNA by serving as decoys [62]. One example is the aforementioned lncRNA PANDA, which interacts with the translation factor NF-YA and prevents its binding to chromatin [59]. Besides, acting as scaffolds, lncRNAs can bring several proteins into discrete complexes [50, 62], such as the previously mentioned lncRNA HOTAIR, which associates with PRC2 and CoREST-LSD complex to specify the correct functions [54, 55]. Moreover, lncRNAs can act as guides to direct the appropriate localization of specific protein complexes [62]. For instance, lncRNA lincRNA-p21 represses transcription through physically association with nuclear factor hnRNP-K and modulating hnRNP-K to proper localization [63].

1.2.2 lncRNA identification

Despite that some lncRNAs have been studied, the functions of most lncRNAs remain unclear. Further characterizing lncRNAs will improve the understanding of transcriptomics. To characterize lncRNAs, two issues are essential to be addressed, lncRNA iden-

tification and functional annotation. lncRNA identification refers to distinguish lncRNAs from protein-coding transcripts (also known as mRNAs), for instance, identifying lncRNAs from the assembled transcripts of a RNA-seq dataset. There are two kinds of methods for lncRNA identification (or assessing the coding potential of transcripts), experimental and computational methods. Usually, experimental methods are used to assess the coding potential of transcripts or test whether a transcript can be translated into proteins [64], which are limited to small-scale. For example, if the function of the transcript is known, researchers can test if the transcript can still exert function when its putative ORFs are perturbed [64, 65]. If the function is unknown, an alternative approach is to test if the transcript can be translated into proteins in vitro, while the drawback is that translation in vitro does not warrant translation in vivo [64]. Moreover, translation does not necessarily indicate function, as some translated peptides can be unstable.

To identify lncRNAs from high-throughput dataset, computational methods, which can identify lncRNAs in large-scale, are preferred than experimental methods. Current computational methods usually integrate multiple features (such as ORF length, nucleotide *kmer* frequency, and conservation features) to asses the coding potential of transcripts [66, 67, 64]. According to the dependence on alignment, lncRNA identification methods can be divided into two categories, alignment-free and alignment-based methods [68]. For instance, CPC [69] and PhyloCSF [70] are alignment-based methods, where CPC requires aligning transcripts to protein databases with BLAST, and PhyloCSF needs multiple sequence alignment to compute the conservation score. CNCI [66], CPAT [67], and PLEK [71] are alignment-free methods. Compared with alignment-free methods, alignment-based methods are less time-effective. From the aspect of relying on reference genome annotation, lncRNA identification methods can be divided into reference-free and reference-based methods. For example, lncRScan-SVM [72] integrates features (such as exon length and count) into a support vector machine (SVM). To extract the features of exon length and count, a reference genome with annotation is required. Reference-based methods can suf-

fer limitations for non-model organisms lacking whole genome sequence or comprehensive gene annotation. A detailed introduction to these computational methods is given in Chapter 2. Although several lncRNA identification methods have been developed, however, identifying lncRNA still remains a challenge, since lncRNA and mRNAs share many similarities, such as transcript length, poly A tails, and splicing structures. Besides, as aforementioned, because of the short length of reads from next-generation sequencing technologies, it is difficult to reconstruct full-length transcripts from RNA-seq dataset [29]. If the reconstructed partial-length mRNAs contain only partial coding sequence, they tend to be misclassified into lncRNAs. In other words, lncRNA identification is further complicated by partial-length transcripts.

1.2.3 lncRNA functional annotation

After identifying lncRNAs, the next step is to annotate their functions. Annotating the functions is not simple, as the functions of lncRNAs are complicated. As aforementioned, lncRNAs can interact with DNAs, RNAs, and proteins, and the lncRNA-protein interactions are confirmed to play crucial roles in the functions of lncRNAs. Therefore, some methods are proposed to infer the functions by investigating lncRNA-protein interactions, including experimental and computational methods. There are several experimental methods, such as RNA immunoprecipitation (RIP), UV cross-linking and immunoprecipitation (CLIP), and Chromatin Isolation by RNA purification (ChIRP) [73]. RIP is a technique to detect protein-RNA interactions in vivo based on the immunoprecipitation of a target protein, where formaldehyde is used to treat live cells and generate protein-RNA cross-links [74]. Several PRC2-interacting lncRNAs were discovered by RIP, such as Xist, Tsix, and RepA [75]. CLIP is also an technique to identify protein-RNA interactions related to RNA immunoprecipitation, however, CLIP uses ultraviolet light (UV) for cross-linking and can give the positional information for binding site [76]. Contrasted to RIP and CLIP, which are antibody-based techniques studying the proteins of interest, ChIRP is a tech-

nique to investigate a certain lncRNA. ChIRP uses antisense biotinylated oligonucleotides, which are complementary to the target lncRNAs, to capture the target lncRNA-protein-chromatin complex for protein analysis or DNA sequencing [77, 78]. For instance, ChIRP was used to investigate the binding sites of HOTAIR on chromatin, and it was observed that the occupancy of HOTAIR did not rely on protein EZH2 "when recruiting PRC2 to its targets genes" [78]. Apart from investigating the interactions between lncRNAs and proteins/RNAs/DNAs, recently, CRISPR was used to control the transcriptional activation or gene expression, which are alternative approaches for studying the functions of lncRNAs [79, 80].

Although experimental methods are promising, they are expensive and time-consuming. In contrast, computational methods are faster and more convenient, which can also provide helpful information for the functions of lncRNAs. Computational methods can be divided into two categories, expression-based and expression-free methods, where expression-based methods require the expression profile of lncRNAs, which are usually from RNA-seq datasets. Several databases and tools used expression-based methods to predict the functions of lncRNAs, such as LncRNA2Function [81], FARNA [82] and IRWRLDA [83]. lncRNA2function inferred the functions of lncRNAs by analyzing the expression correlation between lncRNAs and mRNAs, where the lncRNA-coexpressed mRNAs can be used for functional terms enrichment analysis [81]. FARNA is also based on the coexpression analysis between lncRNAs and transcription factors (and transcription-factor co-factors) [82]. IRWRLDA integrates lncRNA-expression similarity, disease semantic similarity, lncRNA-disease associations into a similarity network and infer novel lncRNA-disease associations [83]. However, expression-based methods can suffer some drawbacks, for example, if few coexpressed mRNAs (or translation factor) of the target lncRNA are located, expression-based methods cannot provide satisfactory functional annotation for the target lncRNA. In contrast, expression-free methods do not rely on the expression profiles and usually predict the interactions between lncRNAs and DNAs/RNAs/proteins, such as

LongTarget [84], lncTar [85], and lncPro [86]. Based on Hoogsteen base-pairing analysis, LongTarget predicts lncRNA-DNA binding sites and motifs [84]. lncTar predicts lncRNA-RNA interactions by computing the normalized free energy of a paired RNAs [85]. lncPro uses the structure information of lncRNAs and proteins to predict their interactions [86]. However, some expression-free methods do not provide executable tools and cannot provide detailed functional annotations, increasing the difficulty for users to apply these methods on novel lncRNAs. In a few words, computational methods can be used to guide the experiments, for example, researchers can conduct CLIP to verify the predicted lncRNA-interacting proteins.

1.3 Neural networks and deep learning methods

Consisting of multiple processing layers for learning representations of data, deep learning methods have improved the state-of-the-art in many domains, such as speech recognition, visual object recognition and bioinformatics [87]. Compared with the shallow machine learning architectures (such as SVM and logistic regression), deep learning methods are better at discovering intricate hidden structures in high-dimensional data, which is particular helpful for classification problems in bioinformatics [88, 89].

1.3.1 Neural networks and back propagation

Deep learning methods are based on multilayer neural networks, which can be trained with back propagation [87, 90]. Here is a brief introduction to a simple neural network and back propagation (Figure 1.2). The neural network in Figure 1.2 is composed of one input layer,

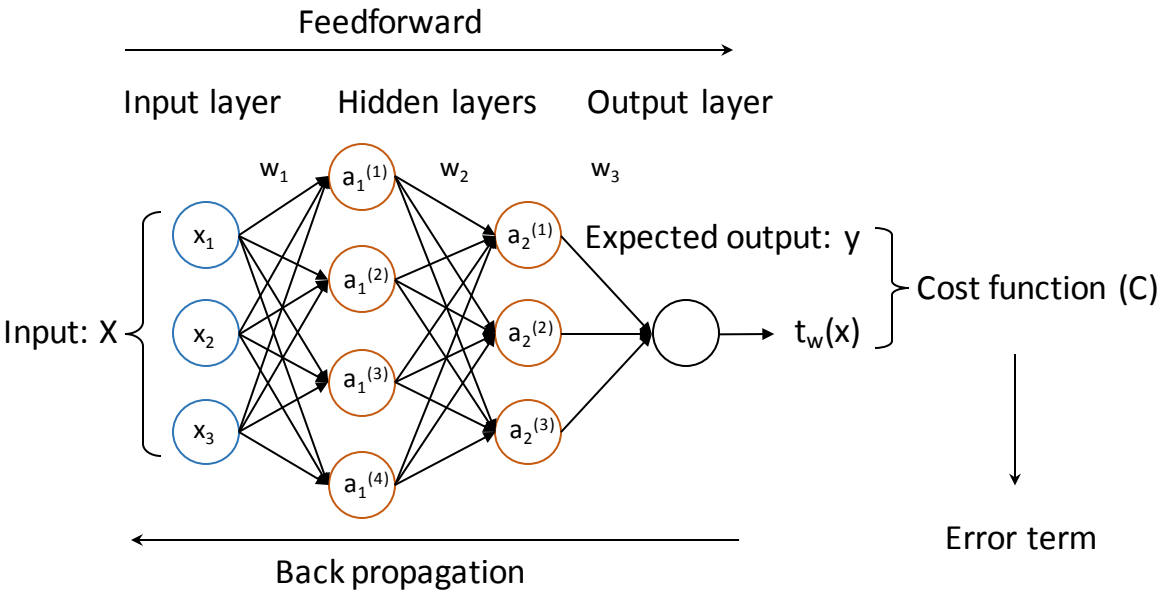


Figure 1.2: The architecture of a simple neural network.

two hidden layers, and one output layer, which are connected with weights,

$$\begin{aligned} \mathbf{a}_1 &= \sigma(\mathbf{W}_1\mathbf{x}) \\ \mathbf{a}_l &= \sigma(\mathbf{W}_l\mathbf{a}_{l-1}) \\ \mathbf{z}_l &= \mathbf{W}_l\mathbf{a}_{l-1} \\ \mathbf{t}_w(x) &= \mathbf{a}_3 \end{aligned} \tag{1.3}$$

where $\mathbf{x}$ refers to the input vector, $\mathbf{W}_l$ represents the weight parameters associated with the layers $l-1$ and l, $\mathbf{a}_l$ denotes the activation vector of layer l, $\mathbf{t}_w(x)$ is the output of the neural network, and σ is the activation function. For simplicity, the bias term is omitted here. Many choices are available for the activation function, such as logistic function, tanh function, and Rectified linear unit (ReLU) [91].

$$\begin{aligned} \text{logistic}(x) &= \frac{1}{1+e^{-x}} \\ \tanh(x) &= \frac{1-e^{-2x}}{1+e^{-2x}} \\ \text{ReLU}(x) &= \max(0,x) \end{aligned} \tag{1.4}$$

After feedforwarding the input ($\mathbf{x}$) to the neural network, the output ($\mathbf{t}_w(x)$) will be generated. To train the neural network, the target is to minimize the discrepancy (cost function) between the output ($\mathbf{t}_w(x)$) and the desired output (y), which can be done with back propagation [90]. Cross-entropy function is a common used cost function for binary classification problem and can be computed as follows:

$$C = -\frac{1}{n}\sum \left[y \ln \mathbf{t}_w(x) + (1-y)\ln(1-\mathbf{t}_w(x))\right], \tag{1.5}$$

where n denotes the total number of input items in training data. To minimize the cost function, gradient descent can be used to update the weight parameters, such that:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta \nabla C(\mathbf{w}^{\tau}) \tag{1.6}$$

where $\mathbf{w}^{\tau}$ refers to the weight parameters at iteration τ, and η is the learning rate. Here, ∇C need be computed for each layer using back propagation. First, consider the last layer L and use the chain rule (suppose using logistic function as activation function). To simplify the notation, let $n = 1$.

$$\begin{aligned} \frac{\partial C}{\partial w_L} &= \frac{\partial C}{\partial t}\frac{\partial t}{\partial z_L}\frac{\partial z_L}{\partial w_L} \\ &= \frac{y-t}{t(1-t)}\sigma_L^{'} a_{L-1} \\ &= (y-t)a_{L-1} \end{aligned} \tag{1.7}$$

As $\frac{\partial z_L}{\partial w_L}$ can be explicitly computed, we only need to focus on $\frac{\partial C}{\partial z_L}$, which is the error term denoted as δ:

$$\delta_L = \frac{\partial C}{\partial z_L} = \frac{\partial C}{\partial t}\frac{\partial t}{\partial z_L} = y - t \tag{1.8}$$

Then, consider the layers $L-1$ and $L-2$,

$$\begin{aligned}
\delta_{L-1} &= \frac{\partial C}{\partial z_{L-1}} = \frac{\partial C}{\partial t}\frac{\partial t}{\partial z_L}\frac{\partial z_L}{\partial a_{L-1}}\frac{\partial a_{L-1}}{\partial z_{L-1}} \\
&= \delta_L w_L \sigma_{L-1}^{'} \\
\delta_{L-2} &= \frac{\partial C}{\partial z_{L-2}} = \frac{\partial C}{\partial t}\frac{\partial t}{\partial z_L}\frac{\partial z_L}{\partial a_{L-1}}\frac{\partial a_{L-1}}{\partial z_{L-1}}\frac{\partial z_{L-1}}{\partial a_{L-2}}\frac{\partial a_{L-2}}{\partial z_{L-2}} \\
&= \delta_{L-1} w_{L-1} \sigma_{L-2}^{'}
\end{aligned} \tag{1.9}$$

The other layers can be computed similarly. Consequently, for layer l and $l-1$,

$$\begin{aligned}
\delta_l &= \delta_{l+1} w_{l+1} \sigma_l^{'} \\
\frac{\partial C}{\partial w_l} &= \delta_l \frac{\partial z_l}{\partial w_l}
\end{aligned} \tag{1.10}$$

The weight parameters can then be updated using Equation 1.6.

As an efficient procedure for multiple-layer neural network training, back propagation has been used in many deep learning architectures, such as deep belief network (back propagation is used to finetune the network) [92], convolutional neural network [93], and recurrent neural network [94]. Various deep learning architectures offer alternative approaches to further improve the classification performance apart from novel models.

1.3.2 Deep belief network

Herein, we give a brief introduction to deep belief network (DBN) [92]. As a generative graph model, DBN was proposed around 2006 by *Hinton, Geoffrey et al*. [92] and worked well on many applications, such as recognizing handwritten digits and speech recognition [92, 95, 96]. A DBN is built as a stack of restricted Boltzmann machines (RBMs) [92, 89], and a RBM (Figure 1.3) is a kind of energy-based model composed of one layer of hidden variables and one layer of observed variables [92, 89, 97].

For the RBM, the joint distribution $P(\mathbf{x}, \mathbf{h})$ over hidden variables $\mathbf{h}$ and observed vari-

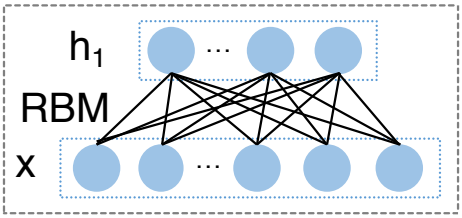


Figure 1.3: The graph model of a RBM.

ables $\mathbf{x}$ is defined with respect to an energy function E,

$$P(\mathbf{x}, \mathbf{h}) = \frac{exp(-E(\mathbf{x}, \mathbf{h}))}{Z} \tag{1.11}$$

where $Z = \sum_{\mathbf{x},\mathbf{h}} exp(-E(\mathbf{x}, \mathbf{h}))$ and the energy function is defined as:

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{b}'\mathbf{x} - \mathbf{c}'\mathbf{h} - \mathbf{h}'\mathbf{W}\mathbf{x} \tag{1.12}$$

The marginal distribution $P(\mathbf{x})$ can be calculated as:

$$P(\mathbf{x}) = \frac{\sum_{\mathbf{h}} exp(-E(\mathbf{x}, \mathbf{h}))}{Z} \tag{1.13}$$

For a Bernoulli (hidden)-Bernoulli (observed) RBM (i.e., $\mathbf{h}_i \in \{0, 1\}, \mathbf{x}_i \in \{0, 1\}$), the conditional probabilities can be computed as:

$$p(\mathbf{h}_i = 1|\mathbf{x}) = \sigma(\mathbf{c}_i + W_i\mathbf{x}), p(\mathbf{x}_j = 1|\mathbf{h}) = \sigma(\mathbf{b}_j + W'_{.j}\mathbf{h}) \tag{1.14}$$

where $\sigma(x) = 1/(1 + exp(x))$ and $W'_{.j}$ denotes the jth column of W. To estimate the parameters θ of $P(\mathbf{x})$, our target is to maximize $P(\mathbf{x})$ with maximum likelihood estimation. Thus, to use gradient descent method, we need compute the gradient of the log likelihood

$\log P(\mathbf{x})$ and reuse the definition of Expectation.

$$\begin{aligned}\frac{\partial log P(\mathbf{x})}{\partial \theta} &= \frac{\partial \left(log \sum_{\mathbf{h}} exp(-E(\mathbf{x},\mathbf{h}) - log \sum_{\mathbf{x},\mathbf{h}} exp(-E(\mathbf{x},\mathbf{h}) \right)}{\partial \theta} \\ &= -\sum_{\mathbf{h}} \frac{exp(-E(\mathbf{x},\mathbf{h})}{\sum_{\mathbf{h}} exp(-E(\mathbf{x},\mathbf{h})} \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial \theta} + \sum_{\mathbf{x},\mathbf{h}} \frac{exp(-E(\mathbf{x},\mathbf{h})}{\sum_{\mathbf{x},\mathbf{h}} exp(-E(\mathbf{x},\mathbf{h})} \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial \theta} \\ &= Expectation_{P(\mathbf{x},\mathbf{h})} \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial \theta} - Expectation_{P(\mathbf{x}|\mathbf{h})} \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial \theta}\end{aligned} \tag{1.15}$$

Considering the weight parameters W,

$$\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial W_{ij}} = h_i x_j \tag{1.16}$$

Therefore, the weight W can be updated with the following rule:

$$\Delta W_{ij} = Expectation_{P(\mathbf{x},\mathbf{h})}(h_i x_j) - Expectation_{P(\mathbf{x}|\mathbf{h})}(h_i x_j) \tag{1.17}$$

The parameters b and c can be updated similarly. To update the parameters, $h_i x_j$ can be calculated through sampling from $P(\mathbf{x},\mathbf{h})$ and $P(\mathbf{x}|\mathbf{h})$. As for a Bernoulli (hidden)-Bernoulli (observed) RBM, the conditional probabilities are analytically available (Equation 1.14), and Monte Carlo Markov Chain (MCMC) sampling [98] with k steps can be used.

$$\begin{aligned}\mathbf{x_1} &\sim \widehat{P}(\mathbf{x}) \\ \mathbf{h_1} &\sim P(\mathbf{h}|\mathbf{x_1}) \\ \mathbf{x_2} &\sim P(\mathbf{x}|\mathbf{h_1}) \\ &\vdots \\ \mathbf{x_{k+1}} &\sim P(\mathbf{x}|\mathbf{h_k})\end{aligned} \tag{1.18}$$

However, MCMC tends to be time-expensive and therefore an approximate sampling strategy known as contrastive divergence (usually one step of sampling is adequate) was pro-

posed, which can be referred to in [92]. Stacking a number of the RBMs learned layer by layer from bottom-up gives rise to a DBN (Figure 1.4), and one immediate advantage of DBN is that even unlabeled dataset can be used for pre-training the neural network and then obtaining a good initialization point [92, 87, 99]. After initialization, an output layer can be added and then the whole neural network can be fine-tuned with back propagation. Pre-training can help to prevent overfitting and is especially helpful for small datasets [87, 99].

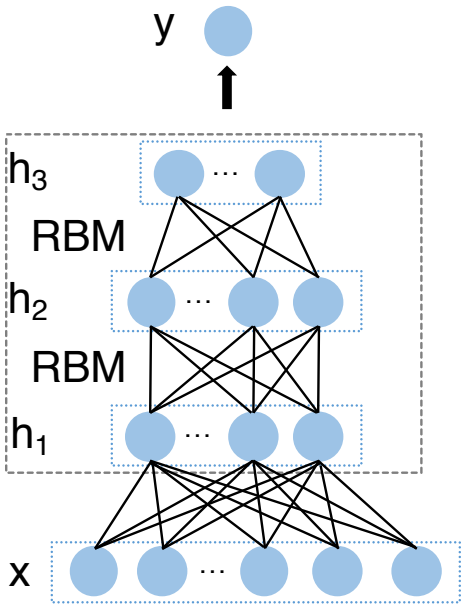


Figure 1.4: The structure of a DBN.

1.4 Introduction to the contents of this dissertation

lncRNAs play important biological roles and have been implicated in disease such as cancer. At present, RNA sequencing has become a prevalent method to study transcriptome, especially lncRNAs. To comprehensively annotate newly discovered transcripts, the first step is to distinguish lncRNAs from mRNAs. Although several lncRNA identification tools are available, most of them focus on identifying lncRNAs from full-length transcripts, while transcripts assembled from RNA-seq data are composed of full- and partial-length, limiting their applications and performance. To exert biological functions, lncRNAs can

interact with DNAs, RNAs, and proteins (lncRNA-protein interactions). Among the three categories of interactions, lncRNA-protein interactions play crucial roles in the functioning of lncRNA, providing details of how lncRNAs exert functions in various biological processes. Although there are several methods for predicting lncRNA-protein interactions, they can not give functional annotations for lncRNAs, except outputting the interaction results. Recently, deep learning methods have improved the state-of-the-art performance in many areas, and it is appealing to employ deep learning methods into bioinformatics, such as characterizing lncRNAs. To meet the urgent demand for studying lncRNAs, in this dissertation, we develop a novel lncRNA identification and functionally annotation tool named LncADeep (http://cqb.pku.edu.cn/ZhuLab/LncADeep), based on deep learning methods. Herein, we give a brief introduction for the contents of this dissertation.

(i) Identify lncRNA with deep learning. First, we develop a novel method for lncRNA identification (implemented as LncADeep), improving the accuracy of identifying lncRNAs. As a widespread method for studying lncRNAs, current RNA-seq techniques are mainly based on next-generation sequencing technologies and tend to generate short reads, which impedes the accurate reconstruction of full-length transcript. Thus, the partial-length mRNAs assembled from RNA-seq short reads further complicate the identification of lncRNAs, as partial-length mRNAs truncated at 5' and/or 3' end can result in incomplete coding sequences, which are much likely to be misclassified into lncRNAs. Moreover, in real dataset consisting of lncRNAs and full- and partial-length mRNAs, it is unclear whether any given transcript is of full- or partial-length, therefore, the composition of full- and partial-length mRNAs is unknown and varies case by case. As the composition of full- and partial- length mRNAs in training dataset can affect the performance of lncRNA identification on test dataset, the classifier trained on the dataset with an arbitrary composition of full- and partial-length mRNAs might not perform well on real dataset. To address these challenges, we propose a novel lncRNA identification method by integrating sequence intrinsic and homology features based on deep belief networks. Particularly, we construct two

models for lncRNA identification, one targeting full-length transcripts and the other targeting transcripts including full- and partial-length. Furthermore, we employ majority voting to avoid using a classifier trained on the dataset with an arbitrary composition of full- and partial-length mRNAs. Based on a comprehensive benchmark with eleven state-of-the-art lncRNA identification tools, CPC [69], CPC2 [100], CPAT [67], CNCI [66], PLEK [71], lncRScan-SVM [72], lncRNA-MFDL [101], COME [102], longdist [103], lncScore [104], and FEElnc [105], our proposed tool LncADeep has outperformed them on full-length human transcripts with an average accuracy of 97.7% (sensitivity 98.1%, specificity 97.2%), and transcripts including full- and partial-length with an average accuracy of 94.2% (sensitivity 93.8%, specificity 94.5%). In addition, LncADeep still achieved better performance than the above tools on cross-species lncRNA identification on full-length mouse transcripts with an accuracy of 96.7% (sensitivity 97.0%, specificity 96.3%) and transcripts including full- and partial-length with an accuracy of 94.2% (sensitivity 95.1%, specificity 93.3%).

(ii) Predict lncRNA-protein interactions and infer lncRNA functions. Second, as lncRNA-protein interactions play the critical roles in the functioning of lncRNAs, we develop a method for predicting lncRNA-protein interactions and then the predicted interacting proteins are used for annotating the functions of lncRNAs (implemented as LncADeep). To predict lncRNA-protein interactions, we integrate sequence and structure features of lncRNAs and proteins, which are fed into deep neural networks to build a binary classifier. Based on a comprehensive benchmark with state-of-the-art tools (i.e., lncPro [86], RPISeq [106], RPI-pred [107], rpiCool [108] and IPMiner [109]), LncADeep has outperformed them with the highest accuracy of 90.8% (sensitivity 97.0%, specificity 85.4%). After predicting the interacting proteins of lncRNAs, LncADeep conducts KEGG and Reactome pathway enrichment analysis and functional module detection to infer the functions of lncRNAs. Therefore, apart from predicting lncRNA-protein interaction with better performance, LncADeep can provide functional annotations for lncRNA automatically, show-

ing noticeable advantages over the above five tools. We then use LncADeep to annotate the 27,384 lncRNAs collected by GENCODE database, and results have shown that LncADeep annotated each lncRNA with an average of 25 KEGG and 67 Reactome pathways, which are more abundant than the annotations resulted from IPMiner's predicted proteins for enrichment analysis. Moreover, for lack of a gold standard dataset for lncRNA functions, we took several well-studied lncRNAs as examples by comparing their inferred functions (by LncADeep) with the reported functions from literatures. Case studies show that LncADeep's functional annotations for lncRNAs conform with their known functions.

In summary, we develop a novel lncRNA identification and functional annotation tool, LncADeep, which integrates the above two proposed methods, i.e., (i) identify lncRNA with deep learning and (ii) predict lncRNA-protein interactions and infer lncRNA functions. LncADeep (http://cqb.pku.edu.cn/ZhuLab/LncADeep/) is freely available for non-commercial use. As an executable tool, LncADeep has two distinctive characteristics. First, LncADeep outperformed state-of-the-art lncRNA identification tools on full-length transcripts and transcripts including full- and partial-length, which is particularly useful for identifying lncRNAs from RNA-seq dataset. Second, LncADeep outperformed state-of-the-art tools on predicting lncRNA-protein interactions and can provide informative functional annotations, which comply with the known functions, for lncRNAs. As far as we know, LncADeep is the first tool which can identify lncRNA and annotate lncRNAs functions automatically. It is expected that LncADeep can not only identify lncRNAs with high performance, but also offer informative functional annotations for studying the associations among lncRNAs, gene regulation and diseases, and further facilitate the large-scale automatic genome annotation.

The rest of this dissertation is organized as follows. Chapter 2 describes the novel method we developed for lncRNA identification. Chapter 3 introduces the method we designed for predicting lncRNA-protein interactions and inferring the functions of lncRNAs.

Each of these chapters gives introduction and motivation, describes materials and methods for the study, and presents results and discussions. In the end, apart from giving an illustration for the application of LncADeep on an RNA-seq dataset, Chapter 4 concludes this dissertation. In addition to developing LncADeep for lncRNA identification and functional annotation, we investigate the impact of RNA-seq aligners on DEG detection and gene expression estimation, since sequence alignment is a key step in RNA-seq data analysis, and the accuracy of alignment will profoundly affect the subsequent analysis. As these are the minor part of this dissertation, we present them in Appendices. Appendix A and B studies the impact of RNA-seq aligners on DEG detection and gene expression estimation, respectively.

CHAPTER 2
IDENTIFY LNCRNA WITH DEEP LEARNING

Long noncoding RNAs play important biological roles and have been implicated in various human diseases. To characterize lncRNAs, identifying and annotating lncRNAs is necessary. This chapter describes the method we proposed for identifying lncRNAs based on a deep learning algorithm. The proposed lncRNA identification method has been implemented into the tool named LncADeep (http://cqb.pku.edu.cn/ZhuLab/LncADeep).

2.1 Introduction

While approximately three quarters of human genome is pervasively transcribed, less than 2% encodes for proteins [47, 48, 49], indicating a large proportion of the genome is transcribed into noncoding RNAs. As the majority of noncoding RNAs, long noncoding RNAs (lncRNAs, length above 200 nt) [47] play important biological roles in dosage compensation, genomic imprinting, cell differentiation [47, 50], and have been implicated in human disease such as cancers [51]. Although some lncRNAs have been characterized, the functions of most lncRNAs currently remain unclear [48].

To comprehensively characterize newly discovered transcripts, two issues are required to be addressed: identifying lncRNAs and inferring their functions. As the first step, lncRNA identification still remains a challenge. First of all, lncRNAs and mRNAs share many similarities such as transcript length and splicing structure [110, 50], which complicate lncRNA identification. Facilitated by high-throughput sequencing technologies, RNA sequencing (RNA-seq) has become a prevalent method for studying lncRNAs [111]. However, accurate full-length transcript assembly is impeded by the short reads from current RNA-seq techniques [29]. For example, according to an assessment, the best-performing unguided assembly methods identified only 21% full-length mRNAs from human datasets,

and over 60% of reconstructed transcripts were of partial-length [29]. Thus, lncRNA identification is further complicated by partial-length mRNAs reconstructed from RNA-seq short reads (Figure 2.1), since partial-length mRNAs truncated at 5' and/or 3' end can lead to incomplete coding sequences (CDS), which are prone to be misclassified as lncRNAs. Moreover, in real dataset consisting of lncRNAs and full- and partial-length mRNAs, it is unclear whether any given transcript is of full- or partial-length, therefore, the composition of full- and partial-length mRNAs is unknown and varies case by case. As the composition of full- and partial-length mRNAs in training dataset can affect the performance of lncRNA identification on test dataset, the classifier trained on the dataset with an arbitrary composition of full- and partial-length mRNAs might not perform well on real dataset.

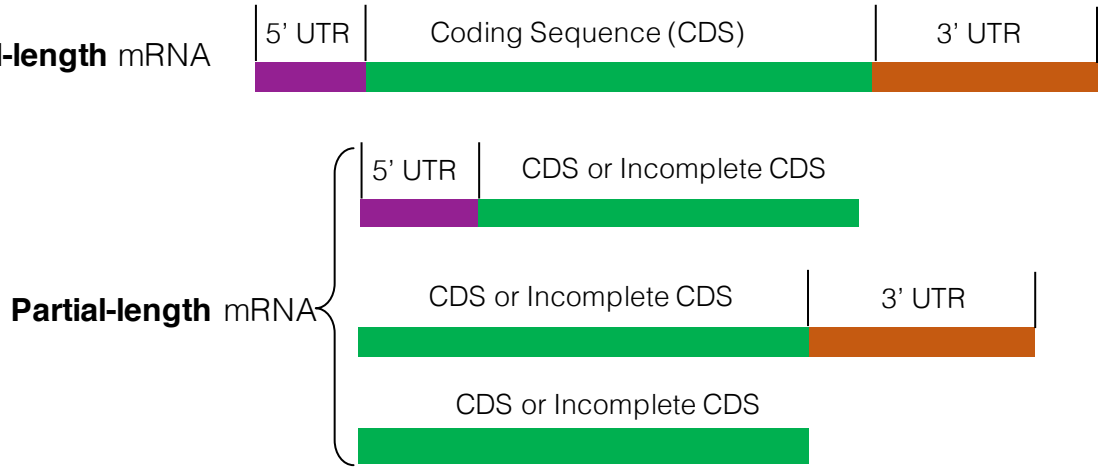


Figure 2.1: The structures of reconstructed full- and partial-length mRNAs.

lncRNA identification methods can be divided into two categories, reference-based and reference-free methods, where reference-based methods require comprehensive reference genome annotation (Table 2.1). lncRScan-SVM [72], COME [102], PhyloCSF [70], and lncScore [104] are reference-based methods. lncRScan-SVM combines features (such as exon length and count and PhastCons score) into a support vector machine (SVM) to classify lncRNAs and mRNAs. To extract the features, such as exon length and count, aligning the transcripts to a reference genome with annotation is necessary. COME [102] integrates sequence-derived, expression, and histone features into a Random Forest classifier

Table 2.1: Tools for lncRNA identification

Tool	Algorithm	Reference-Based	Year	Journal
CPC	SVM	No	2007	Nucleic Acids Res
CPC2	SVM	No	2017	Nucleic Acids Res
PhyloCSF*	Phylogenetic model	Yes	2011	Bioinformatics
CPAT	Logistic Regression	No	2013	Nucleic Acids Res
CNCI	SVM	No	2013	Nucleic Acids Res
PLEK	SVM	No	2014	BMC Bioinformatics
lncRNA-ID*	Random forest	No	2015	Bioinformatics
lncRNA-MFDL	Deep stacking networks	No	2015	Molecular Biosystem
lncRScan-SVM	SVM	Yes	2015	PLOS one
COME	Random forest	Yes	2016	Nucleic Acids Res
lncScore	Logistic Regression	Yes	2016	Scientific Reports
FEElnc	Random forest	No	2017	Nucleic Acids Res
longdist	SVM	No	2017	BMC Genomics
LncADeep	Deep belief networks	No	2017	This dissertation

for lncRNA identification, requiring genome annotation and experimental datasets to index the reference genome and to align the transcript to reference genome. PhyloCSF [70] applies multiple sequence alignment, which requires reference genome annotation, to calculate phylogenetic conservation score. lncScore [104] employs features (i.e., features from exon, maximum coding subsequence, and ORF) into logistical regression, and also considers the existence of partial-length mRNAs assembled from RNA-seq data. To extract features from exons, lncScore requires aligning transcripts to the reference genome with annotation. Reference-based methods can suffer limitations for non-model organisms lacking whole-genome sequence or comprehensive reference gene annotation [66, 68]. Even though the reference genome and genome annotation are available, the potential multiple-mapping of a transcript can complicate the feature extraction, thus affecting the subsequent lncRNA identification.

CPC [69], lncRNA-ID [68], CPAT [67], CPC2 [100], CNCI [66], PLEK [71], FEElnc [105], longdist [103], and lncRNA-MFDL [101] are reference-free methods. CPC [69]

employs BLAST to align transcripts to known protein databases, with an underlying assumption that noncoding transcripts tend to share less sequence conservation than homologous mRNAs. lncRNA-ID [68] takes advantage of profile hidden Markov model-based alignment [112], however, lncRNA-ID does not provide an executable tool, limiting its application. CPAT [67] incorporates linguistic features of transcript sequence into a logistic regression model to assess coding potential. As an updated version of CPC [69], CPC2 [100] employs sequence intrinsic features with SVM. CNCI [66] distinguishes lncRNAs from mRNAs by profiling adjoining nucleotide triplets with SVM. Longdist [103] employs PCA to select k-mer based features and identifies lncRNAs using SVM. PLEK [71] classifies lncRNAs and mRNAs with SVM using *k-mer* based features. FEElnc [105] employs *k-mer* frequencies and relaxed open reading frames to predict lncRNA with Random forests. lncRNA-MFDL [101] identifies lncRNAs by integrating linguistic features and predicted secondary structures into deep stacking networks. However, RNA structure prediction is error-prone, computational expensive, and the accuracy is often unsatisfactory for RNAs longer than 1000 nt [113] , and might introduce unexpected effects to the subsequent lncRNA identification. Besides, predicting secondary structure is computational expensive: the time complexity of RNAfold [113] used by lncRNA-MFDL is $O(L^3)$, where L is the length of RNA sequence.

Most of the above methods focus on only full-length transcripts. Although CNCI, lncScore and FEElnc realized the existence of partial-length mRNAs, they did not consider the various compositions of full- and partial-length mRNAs in real dataset and used only an arbitrary composition of full- and partial-length mRNAs for training, which could affect the performance of lncRNA identification.

With the development of high throughput sequencing technology, a large amount of transcripts will be sequenced and require functional annotation. To fulfill the requirements, we present LncADeep, which can not only identify lncRNAs, but also infer the functions of lncRNAs, while no tools can accomplish both. This chapter focuses on lncRNA identifica-

tion, where LncADeep integrates sequence intrinsic features (e.g., entropy density profile) and homology features (i.e., profile hidden Markov model-based alignment) into a deep belief network, improving the accuracy of identifying lncRNAs. Herein, we construct two models for lncRNA identification, one targeting full-length transcripts and the other targeting transcripts including full- and partial-length. Identifying lncRNAs from full-length transcripts will be useful when the third-generation sequencing technologies, which suffice generating full-length transcripts, are prevalent. Moreover, we propose to use majority voting to avoid using a classifier trained on the dataset with an arbitrary composition of full- and partial-length mRNAs. To our knowledge, LncADeep is the first tool considering the various compositions of full- and partial-length mRNAs in dataset. LncADeep has outperformed state-of-the-art tools including CNCI, COME, CPC, CPC2, CPAT, FEElnc, lncRNA-MFDL, longdist, lncRScan-SVM, lncScore and PLEK on full-length transcripts, transcripts including full- and partial-length, and cross-species lncRNA identification.

2.2 Materials and methods

2.2.1 Data description

Currently both RefSeq [114] and GENCODE [48, 115] provide comprehensive and well-annotated sequences for mRNAs and lncRNAs. In particular, the human mRNAs in RefSeq are of full-length. However there are about 36% of human mRNAs in GENCODE (Release 24) are of partial-length, which are mainly owning to the challenges of transcript assembling from high-throughput sequenced RNA-seq data [115]. Herein, full-length mRNAs contains 5' untranslated region (UTR), CDS, and 3' UTR, while partial-length mRNAs can miss 5' UTR or 3' UTR, and the CDS can also be incomplete (Figure 2.1).

As aforementioned, we constructed two models for lncRNA identification, one targeting full-length transcripts and another targeting transcripts including both full- and partial-length ones. In particular, full-length transcripts are composed of lncRNAs and full-length mRNAs, and transcripts including full- and partial-length consist of lncRNAs and full- and

Table 2.2: The parameters of CD-HIT

Option	Value
-c	0.95
-n	10
-M	16000
-T	8
-s	0.95
-S	10
-aL	0.9
-aS	0.9

partial-length mRNAs. Full-length human mRNAs were collected from RefSeq Release 75, while mRNAs including full- and partial-length and lncRNAs were from GENCODE Release 24. To remove similar and redundant transcripts, we used CD-HIT [116] to keep only the representative ones (Table 2.2). After removing short transcripts (length less than 200 nt), we finally obtained 38,679 mRNAs from RefSeq (all full-length), and 91,752 mRNAs (including 58,528 full-length and 33,224 partial-length) and 27,384 lncRNAs from GENCODE (Table 2.3). To assess the performance of cross-species lncRNA identification, we collected transcripts for mouse, as its experimentally verified mRNAs and lncRNAs are more abundant compared with those of other species. Herein 29,739 full-length mouse mRNAs were collected from RefSeq Release 75, while 56,744 mRNAs (39,079 full-length and 17,665 partial-length) and 12,529 lncRNAs were collected from GENCODE Release M9 (Table 2.4).

Table 2.3: Human dataset for lncRNA identification

Type	**Source**	**Number**
lncRNA	GENCODE Release 24	27,384
Full- and partial-length mRNA	GENCODE Release 24	91,752 (58,528 full-length and 33,224 partial-length)
Full-length mRNA	RefSeq Release 75	38,679

Table 2.4: Mouse dataset for lncRNA identification

Type	Source	Number
lncRNA	GENCODE Release M9	12,529
Full- and partial-length mRNA	GENCODE Release M9	56,744 (39,079 full-length and 17,665 partial-length)
Full-length mRNA	RefSeq Release 75	29,739

2.2.2 Methods

To identify lncRNA, we integrate several sequence and homology features as predictor variables into a deep belief network of deep learning architecture and construct two models shown in Figure 2.2. For the model targeting full-length transcripts, we use the following features including ORF length and coverage, the EDP of ORF, Mean hexamer score, UTR coverage and GC content, Fickett nucleotide feature, and HMMER index.

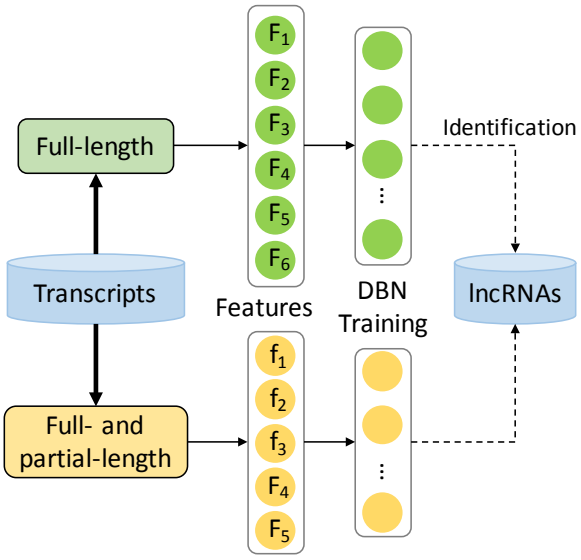


Figure 2.2: The flowchart of lncRNA identification.

We construct two models for lncRNA identification, one for full-length transcripts, and the other for transcripts including full- and partial-length. F_1 to F_6 refer to ORF length and coverage, the EDP of ORF, Mean hexamer score, Fickett nucleotide features, HMMER index, and UTR length and GC content, respectively. f_1 to f_3 represent LCDS length and coverage, the EDP of LCDS, and Mean hexamer score, respectively.

ORF length and coverage

Open reading frame (ORF) is one of the most fundamental criteria to distinguish lncRNAs from mRNAs, since lncRNAs tend to have shorter ORFs or lower ORF proportion than mRNAs [67, 68]. Therefore, we use two features, ORF length and coverage. The ORF length is defined as the length of the longest ORF (LORF) detected in three forward frames, starting with a start codon (ATG) and ending with a stop codon (TAG, TAA, or TGA). The ORF coverage is the ratio of the ORF length to the transcript length.

The EDP of ORF

In our previous studies of prokaryotic gene prediction, the entropy density profile (EDP) model was used to assess the coding potential of ORFs based on the amino acid composition [117, 118]. The success of EDP model supports the hypothesis that protein-coding and noncoding ORFs distribute separately in the EDP phase space, which might be caused by different selection pressure during evolution [117, 118]. Similarly, we extend the EDP model to lncRNA prediction by computing the EDP of amino acid composition of the ORF. The EDP of amino acid composition of an ORF is defined as [117]:

$$s_i = -\frac{1}{H} p_i \log p_i \tag{2.1}$$

where $H = -\sum_{j=1}^{20} p_j \log p_j$ is the Shannon entropy, p_j is the abundance of *j*th amino acid, and $i, j = 1, \ldots, 20$ represents the index of the twenty amino acids.

k-mer frequency has been used to characterize transcripts in several lncRNA identification methods [67, 71, 101] and proved to be useful. Thus, apart from using the EDP of amino acid composition, we propose to use the EDP of *k-mer* of the ORF as another feature. To be specific, we choose $k = 2$ and obtain sixteen *2-mer* patterns (e.g., AA, AC, AG, AT, etc.). Similar to the EDP of amino acid composition of an ORF, the EDP of *2-mer*

of an ORF is defined as:

$$u_i = -\frac{1}{H} c_i \log c_i \tag{2.2}$$

where $H = -\sum_{j=1}^{16} c_j \log c_j$ is the Shannon entropy, c_j is the abundance of *i*th *2-mer*, and $i, j = 1, \ldots, 16$ represents the index of the sixteen *2-mer*. It is worth noting that *the EDP of ORF* is computed on the LORF.

Mean hexamer score.

Hexamer usage is another important feature to distinguish lncRNAs from mRNAs, which has been demonstrated in [119, 67]. Therefore, we extract the hexamer usage bias of the LORF as another feature. For a given hexamer sequence $S = H_1, H_2, ..., H_m$, we follow the definition of mean hexamer score as [67], which is the mean of the logarithm of the ratio of sequence probabilities under coding and noncoding models:

$$\begin{aligned} \mu(S) &= \frac{1}{m} \log \frac{P_{\text{c}}(S)}{P_{\text{nc}}(S)} \\ &= \frac{1}{m} \log \frac{F_{\text{c}}(H_1) \ldots F_{\text{c}}(H_m)}{F_{\text{nc}}(H_1) \ldots F_{\text{nc}}(H_m)} \\ &= \frac{1}{m} \sum_{i=1}^{m} \log \frac{F_{\text{c}}(H_i)}{F_{\text{nc}}(H_i)} \end{aligned} \tag{2.3}$$

where $P_{\text{c}}(S)$ and $P_{\text{nc}}(S)$ represent the probability of the sequence calculated under the model of coding RNA and noncoding RNA respectively. To compute these probabilities, the hexamer frequency for 4,096 ($4^6 = 4096$) kinds of hexamers are required, thus $F_{\text{c}}(h_j)$ and $F_{\text{nc}}(h_j)$ refer to in-frame hexamer frequency ($j = 1, 2, \ldots, 4096$), computed from coding and noncoding training datasets respectively. In particular, the coding and noncoding hexamer frequency were calculated in 33,957 CDSs (extracted from RefSeq human mRNAs, and we removed redundant CDSs) and 27,384 lncRNAs (from GENCODE), respectively. Generally, protein-coding transcripts tend to show a positive hexamer score,

and lncRNAs negative.

UTR coverage and GC content.

A typical eukaryotic full-length mRNA is composed of a 5' untranslated region (5' UTR), a coding region, and a 3' untranslated region (3' UTR) [120]. To distinguish protein-coding and noncoding transcripts, apart from focusing on assessing the coding potential of ORFs, considering the untranslated regions of transcripts is necessary. For eukaryotic mRNAs, 5' UTRs and 3' UTRs show several noteworthy characteristics. To name a few, 3' UTRs are generally much longer than 5' UTRs; the mean GC content of 3' UTR sequences is smaller than that of 5' UTR [120]. Although lncRNAs can have putative ORFs, it can be rare for lncRNAs to still contain both putative 5' and 3' UTRs which conform with the above characteristics. To capture more intrinsic characteristics of transcripts, we incorporate the features of untranslated regions. Being aware of the discrepancy of length and GC content between 5' and 3' UTRs [120], we calculate the 5' UTR and 3' UTR coverage (i.e., the ratio of the length of 5' or 3' UTR to the length of transcript sequence) and GC content to characterize UTRs. To simplify the model, after detecting the LORF of a transcript, we treat the upstream sequence of start codon as 5' UTR, and the downstream sequence of stop codon as 3' UTR.

Fickett nucleotide features.

Fickett nucleotide features were proposed in [121], including nucleotide composition and position frequency. The nucleotide composition is the percentage of each base contributing to the sequence, as coding sequences tend to have higher GC content. The nucleotide position frequency assesses the degree to which each base is favored in one codon position

over another [121], and is defined as follows:

$$
\begin{aligned}
A_1 &= \text{Number of A's in position 1, 4, 7, } \ldots \\
A_2 &= \text{Number of A's in position 2, 5, 8, } \ldots \\
A_3 &= \text{Number of A's in position 3, 6, 9, } \ldots \\
A_{pos} &= \frac{\text{MAX}(A_1, A_2, A_3)}{\text{MIN}(A_1, A_2, A_3) + 1}
\end{aligned}
\tag{2.4}
$$

C_{pos}, G_{pos}, and T_{pos} are computed similarly. To distinguish protein-coding and noncoding sequence, Fickett provided a lookup table to transform these eight values to a TESTCODE score. However, the lookup table in [121] was generated from 321 coding and 249 non-coding sequences then. Compared with the datasets we collected, TESTCODE score from that small data might not be adequate to characterize sequence features. Therefore, instead of directly using TESTCODE score, we compute the nucleotide composition and position frequency on the whole transcript.

HMMER index.

Since mRNAs are more conserved against characterized proteins than lncRNAs, homology features, such as features derived from the output of BLAST [69] and HMMER search [68], will also benefit lncRNA identification. Although protein sequence databases provide a wealth of resources for conservation search using BLAST, the huge search space leads to high time cost. In contrast, using HMMER [112] search against protein families, such as Pfam [122], has several advantages. For instance, compared with the number of protein sequences, the number of gene families is much smaller, indicating a significantly smaller search space and lower time cost. Therefore, we extract features from the output of HMMER search. In particular, we align each transcript to Pfam Release 29.0 [122] using HMMER [112]. Here, the cutoff of E-value is 0.1, and if more than one alignment is reported, the alignment with the best E-value will be used. Since mRNAs are more likely to

have higher alignment score and longer alignment region than lncRNAs, we extract three features, the alignment score, the ratio of aligned region to the query sequence, and the ratio of aligned region to the profile, from the chosen alignment for a transcript.

The above features (i.e., ORF length and coverage, the EDP of ORF, Mean hexamer score, UTR coverage and GC content, Fickett nucleotide feature, and HMMER index) are for the model of full-length transcripts.

Longest CDS.

For the model targeting transcripts including both full- and partial-length, we introduce a feature of the longest CDS (LCDS).

Currently the RNA-seq data tends to produce transcripts mixed with a significant amount of partial-length mRNAs, which may be easily misclassified into lncRNAs and challenge the performance of lncRNA identification. Considering the existence of partial-length mRNA, we first define the ORF-based CDS representing the longest ORFs with their 3' ends missed, i.e., we search for ORFs starting with a start codon, but lacking an in-frame stop codon. Once all the potential ORFs are detected, we choose the longest one as an ORF-based-CDS.

Second, considering hexamer usage bias in CDS, we search for the candidate CDS with the maximum hexamer score, which is similar to the approach in CNCI to find MLCDS with the highest ANT score [66]. The assumption is that the sequence with higher hexamer score is more likely to be CDS. For a given single hexamer, H, the hexamer score $\lambda(H)$ is computed as follows:

$$\lambda(H) = \log \frac{F_{\text{c}}(H)}{F_{\text{nc}}(H)} \tag{2.5}$$

where $F_{\text{c}}(h_j)$ and $F_{\text{nc}}(h_j)$ ($j = 1, 2, \ldots, 4096$) represent in-frame coding and noncoding hexamer frequency, respectively. Thus, for a given hexamer sequence $S = H_1, H_2, ..., H_m$,

the total hexamer score $\lambda(S)$ can be computed as follows:

$$\lambda(S) = \sum_{i=1}^{m} \log \frac{F_{\text{c}}(H_i)}{F_{\text{nc}}(H_i)} \tag{2.6}$$

Given a transcript, we use a dynamic programming method named Maximum Subarray Sum (MSS) [123] to find a Hexamer-based-CDS. This method is to find a continuous in-frame sub-hexamer sequence, which yields the maximum hexamer score (Figure 2.3). To be specific, we consider the three forward in-frame hexamer sequences for a given transcript, and among the three hexamer sequences, we use MSS to find a sub-hexamer sequence with the largest hexamer score, which is treated as Hexamer-based-CDS.

Finally, we choose the longer one as the longest CDS (LCDS) from the ORF-based-CDS and the Hexamer-based-CDS, with a rationale that the LCDS has the highest coverage over the sequence and more likely to cover the true CDS. With the LCDS, we compute its length and coverage as features, where the coverage is the ratio of the LCDS length to the transcript length.

In addition to the LCDS length and coverage, we also use other features, including Mean hexamer score, the EDP of LCDS, Fickett nucleotide feature, and HMMER index. Among these features, Fickett nucleotide feature and HMMER index are computed the same as in the model for full-length transcripts, while Mean hexamer score and the EDP of LCDS are computed on the LCDS. We do not include the UTR feature, as partial-length transcripts might lack 5' or 3' ends.

2.2.3 Deep belief network

The extracted features from each transcripts can be concatenated into a vector and then fed to a deep learning method for training and testing. Compared with the shallow machine learning architectures (such as SVM and logistic regression), deep learning methods are better at discovering intricate hidden structures in high-dimensional data, which is partic-

Example sequence

CTGCACCAAGGCCGCCAGCGCGAAGAAGGCGGCCAT

ID	Hexamer	Hexamer Score
H1	CTGCAC	0.34
H2	CACCAA	-0.66
H3	CAAGGC	-0.27
H4	GGCCGC	0.65
H5	CGCCAG	0.84
H6	CAGCGC	0.77
H7	CGCGAA	-0.09
H8	GAAGAA	0.82
H9	GAAGGC	0.27
H10	GGCGGC	0.81
H11	GGCCAT	-0.24

2.26

1.9

4.07

Figure 2.3: A toy example for finding a continuous in-frame sub-hexamer sequence yielding the maximum hexamer score.

As a toy example, here we only consider the first forward in-frame Hexamer sequences. We need to find a continuous in-frame sub-hexamer sequence, such that the sub-hexamer sequence has the highest hexamer score. Choosing H4,H5,H6, the hexamer score will be 2.26; choosing H8,H9,H10, the hexamer score will be 1.9. If we choose H4,H5,,H10, we can get the highest hexamer score with 4.07.

ular helpful for classification problems [89, 88]. Many deep learning architectures have been available, such as deep belief network (DBN), deep neural network, convolutional neural network, and recurrent neural network [92, 87]. Various architectures offer alternative approaches to further improve the classification performance apart from novel models. In this dissertation, we implement a DBN to identify lncRNAs. A DBN is a class of deep neural network composed of multiple layers of latent variables, with connections between the layers but not between units within each layer [87]. To implement, a DBN is built as a stack of restricted Boltzmann machines (RBMs), and a RBM consists of one layer of hidden units and one layer of observed units [92]. Stacking a number of the RBMs learned layer by layer from bottom-up gives rise to a DBN. DBN can make use of the dataset for pre-training and obtain a good initialization point for neural network. After initialization, an output layer can be added and the whole neural network can be fine-tuned with back

propagation. We construct DBN with the setting as suggested in [92, 124]: For the first two layers, we use Gaussian (visible)-Bernoulli (hidden) RBM; while for the other layers, we utilize Binary-Binary RBM. The architecture of deep belief network is list in Tables 2.5 and 2.6.

We use DBN to initialize the parameters of neural network and then fine-tune the neural network with back propagation. The dimensions of the features are: 2 (ORF length and coverage) + 36 (The EDP of ORF) + 1 (Mean hexamer score) + 4 (UTR length and coverage) + 8 (Fickett nucleotide features) + 3 (HMMER index) = 54 dimensions (Table 2.5). In the model for transcripts including full- and partial-length, we do not use the UTR length and coverage (4 dimensions). Thus, the input dimension is 50 (Table 2.6).

Table 2.5: The architecture of the DNN for full-length transcripts

Layer	**Nodes Number**	**Activation Function**
Input layer:	54 nodes	
1st hidden layer:	10 nodes	
2nd hidden layer:	10 nodes	Sigmoid
3rd hidden layer:	10 nodes	
Output layer:	1 node	

Table 2.6: The architecture of the DNN for transcripts include full- and partial-length

Layer	**Nodes Number**	**Activation Function**
Input layer:	50 nodes	
1st hidden layer:	10 nodes	
2nd hidden layer:	10 nodes	Sigmoid
3rd hidden layer:	10 nodes	
Output layer:	1 node	

2.2.4 Evaluation metrics

To evaluate the prediction performance of the LncADeep and other existing tools, we use two independent quantities, Sn (sensitivity) and Sp (specificity), and a balanced quantity

Hm (the harmonic mean value of sensitivity and specificity), they are defined as follows:

$$\begin{aligned} \text{Sn} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{Sp} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Hm} &= \frac{2 \times \text{Sn} \times \text{Sp}}{\text{Sn} + \text{Sp}} \end{aligned} \tag{2.7}$$

where TP, TN, FP, and FN represents true positive, true negative, false positive, and false negative respectively. Herein Sn measures the ratio of actual positives that are correctly identified, Sp measures the ratio of true positives in all predicted positives, and Hm is a composite measure of sensitivity and specificity [118]. For lncRNA identification, lncRNAs are regarded as the positive class while protein-coding transcripts as the negative class.

2.3 Results

2.3.1 Performance of lncRNA identification

As a comprehensive annotation tool for lncRNA, LncADeep first aims to accurately identify lncRNAs from newly sequenced transcripts. In this subsection, we report the lncRNA identification accuracies of our LncADeep program, and the performance comparison with current representative tools providing executable programs, CPC [69], CPC2 [100], CPAT [67], CNCI [66], PLEK [71], longdist [103], lncRScan-SVM [72], lncRNA-MFDL [101], COME [102], lncScore [104], and FEElnc [105]. PhyloCSF [70] was not included since it is time-consuming and has been outperformed by CPAT and CNCI [67, 66]. To benchmark these tools, it is fair to re-train all programs on the same training set and test against the same test set. Among the above tools, there are COME, CPAT, lncRScan-SVM, longdist, PLEK and FEElnc (FEElnc requires retraining) providing model-training options, while longdist and PLEK's retraining is very time-consuming as mentioned in users' manual so we did not retrain these two programs. For the tools without any available re-training option, namely CPC, CPC2, CNCI, lncRNA-MFDL and lncScore, we can only use their

pre-built models.

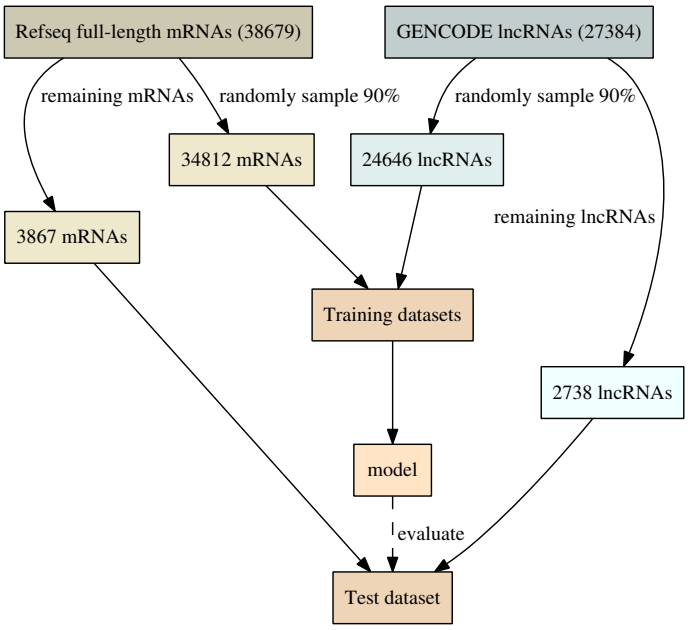


Figure 2.4: The construction of training and test sets for full-length human transcripts.

We first compared the performance of lncRNA identification on full-length transcripts. With 10-fold cross validation (Figure 2.4), LncADeep achieved an average sensitivity of 98.1% and specificity of 97.2%, meanwhile an average harmonic mean of 97.7%. Compared with other tools with the same test sets, LncADeep had the highest specificity, and rather a high sensitivity only slightly lower than that of CPC (however CPC's specificity as 84.8% is much lower than LncADeep's as 97.2%, leading to the harmonic mean of CPC as 91.3% evidently lower than LncADeep's 97.7%). Totally, our method had the highest accuracy among all lncRNA identification tools, and certainly outperformed the existing tools. The comparison of identification results by LncADeep and other tools are shown in Table 2.7. Figure 2.5 plots the mean ROC curves of all tools, and LncADeep achieves the highest AUC of 99.71% and outperforms other tools, which is consistent with the results in Table 2.7. Among the tools compared with in the current study, it should be pointed out that the tools, CNCI, CPC, CPC2, PLEK, lncScore and lncRNA-MFDL, have not been retrained in the process of 10-fold cross validation. So their identification performances might be over

Table 2.7: Comparison of performances for lncRNA identification with 10-fold cross-validation on full-length human transcripts.

Category	Tool	Full-length transcripts[a]		
		Sn (%)	Sp (%)	Hm (%)
Reference-based	COME	94.3±0.3	96.0±0.3	95.1±0.2
	lncScore	93.7±0.5	94.3±0.6	94.0±0.3
	lncRScan-SVM	96.5±0.2	93.1±0.4	94.8±0.3
Reference-free	CPC	**98.9±0.1**	84.8±0.6	91.3±0.3
	CNCI	97.3±0.2	88.2±0.5	92.6±0.2
	CPAT	95.4±0.4	93.6±0.5	94.5±0.3
	CPC2	94.4±0.5	92.4±0.5	93.4±0.4
	FEElnc	96.7±0.3	95.5±0.4	96.1±0.3
	PLEK	98.1±0.2	95.5±0.2	96.8±0.2
	longdist	98.6±0.2	44.5±0.2	61.3±0.2
	lncRNA-MFDL	93.9±0.5	94.8±0.5	94.3±0.4
	LncADeep	98.1±0.2	**97.2±0.3**	**97.7±0.2**

[a] The dataset is from Refseq (full-length mRNAs) and GENCODE (lncRNAs). LncADeep was trained for the model targeting full-length transcripts.

estimated since their training sets have overlap with the test sets. In particular, PLEK's [71] pre-built model also used mRNAs from Refseq and lncRNAs from GENCODE for training, which overlapped with the datasets we used, and this is why PLEK achieved relatively high performance although lower than that of LncADeep.

We then report the identification performances of LncADeep applied on the transcripts including partial-length mRNAs, which certainly has utility value for processing the real data. Our analysis has demonstrated that the ratio of partial-length mRNAs in training set affects the lncRNA identification performance on the same test set (Figure 2.7). So it is more reasonable for the algorithm to train a classifier for a given ratio of partial-length mRNAs within training set as well as introduce the mathematical model of partial-length transcripts as mentioned in section Materials and Methods. To this end, we further designed a majority voting strategy as follows. The LncADeep model was first trained on training sets with various ratios of partial-length mRNAs. Herein we trained 21 classifiers, and each with the percentage of full-length mRNAs from 0% to 100% while partial-length mRNAs

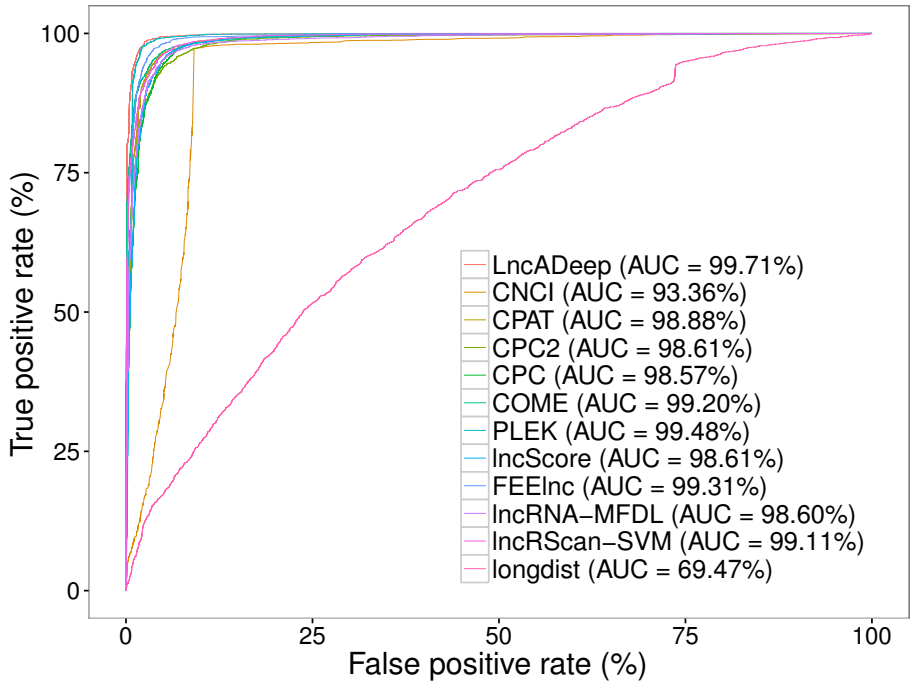


Figure 2.5: Mean ROC curves for lncRNA identification with 10-fold cross-validation on full-length human transcripts.

We plot the mean ROC curves for lncRNA identification by LncADeep and other tools with 10-fold cross-validation on full-length human transcripts. Herein LncADeep achieves the highest AUC of 99.71% and outperforms other tools, which is consistent with the results in Table 2.7.

from 100% to 0% both with a step of 5%. The details for construction of training sets for the 21 classifiers can be referred to Figure 2.6. When using these 21 classifiers for lncRNA identification, each classifier gives an output, then the decision will be voted by 21 outputs. Our results showed that majority voting strategy overall outperformed a single classifier trained on a specific ratio of partial-length mRNAs (Figure 2.7).

To evaluate the lncRNA identification performance of majority voting and various classifiers on transcripts including various compositions of full- and partial-length mRNAs, we constructed various test datasets by randomly combining full- and partial-length mRNAs, where full- and partial-length mRNAs are from the remaining test dataset of every rotation (Figure 2.6), namely, the test dataset had no overlap with training dataset. In Figure 2.7, the legend refers to the classifier trained on transcripts including a specific composition of full- and partial-length mRNAs. For example, Classifier 30_70 represents the classifier trained

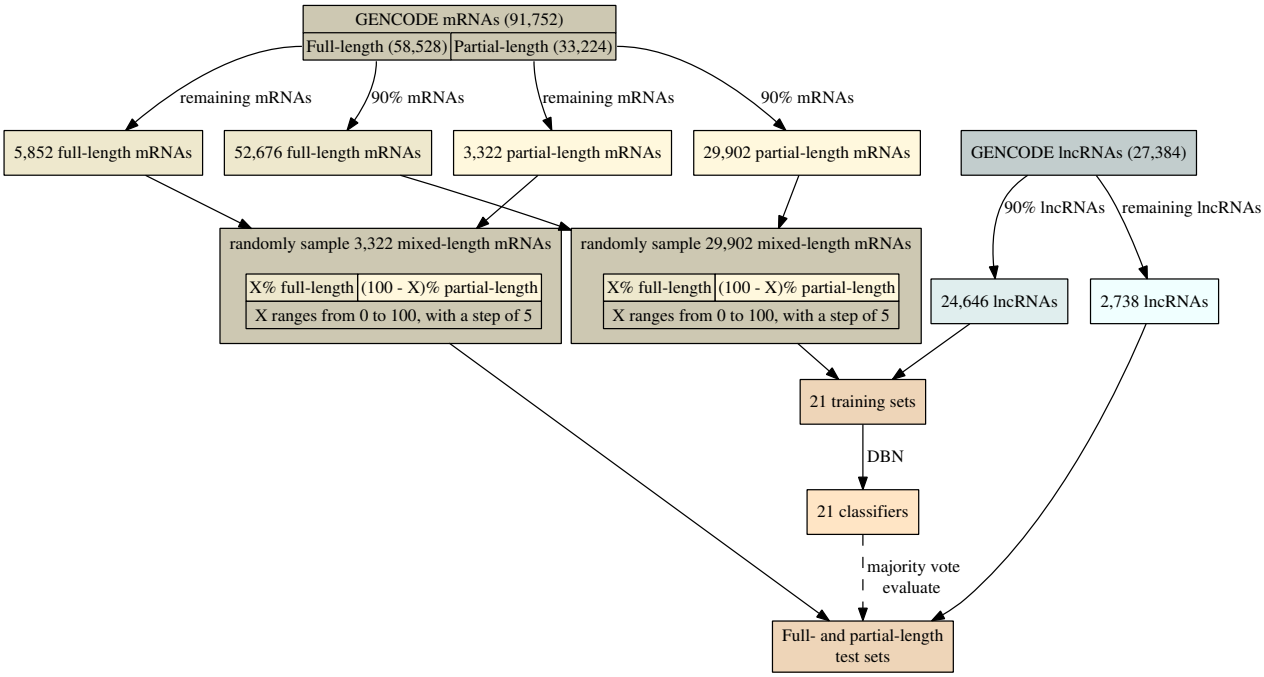


Figure 2.6: The construction of training and test sets for human transcripts with various compositions of full- and partial-length mRNAs.

on transcripts including 30% full-length mRNAs and 70% partial-length mRNAs (the percentage is calculated within the mRNAs). As we can see, various classifiers show different performance on various test datasets, demonstrating that the composition of full- and partial-length mRNAs in training dataset can affect the lncRNA identification performance on the test dataset. For instance, Classifier 0_100 outperformed other classifiers except majority voting on the test dataset with mRNAs composition of 0(full)_100(partial)%, but performed the worst on the test dataset with mRNAs composition of 100(full)_0(partial)%, suggesting that it is not appropriate to use an arbitrary classifier for lncRNA identification in real dataset. In contrast, majority voting consistently outperformed other classifiers, except for the test datasets with mRNA compositions of 95(full)_5(partial)% and 100(full)_0(partial)%. However, the Classifiers 100_0 and 95_5 that outperformed majority voting (on test datasets with mRNAs compositions of 95_5% and 100_5%) performed poorly on the other compositions (compared with majority voting). To keep the generalization performance, it would be better to use majority voting for test dataset whose compo-

sition of full- and partial-length mRNAs is unknown. For the test dataset that is composed of full-length transcripts, we recommend using the model designed for full-length transcripts. Results showed that majority voting overall outperformed a single classifier that was trained on a specific composition of full- and partial-length mRNAs, demonstrating its effectiveness (Figure 2.7).

Table 2.8: Comparison of performances for lncRNA identification with 10-fold cross-validation on full- and partial-length human transcripts.

Tool	Full- and partial-length transcripts[a]					
	Full- and partial-length[b]			Partial-length		
	Sn (%)	Sp (%)	Hm (%)	Sn (%)	Sp (%)	Hm (%)
COME	86.9±0.6	**97.2±0.3**	91.8±0.3	86.9±0.6	**96.1±0.4**	91.2±0.3
lncScore	93.7±0.6	89.7±0.6	91.6±0.3	93.7±0.6	83.5±0.5	88.3±0.4
lncRScan-SVM	81.8±0.8	94.9±0.4	87.9±0.4	81.8±0.8	91.6±0.3	86.4±0.5
CPC	**98.9±0.1**	69.0±0.4	81.3±0.3	**98.9±0.1**	57.5±0.3	72.7±0.3
CNCI	97.3±0.2	79.3±0.7	87.4±0.5	97.3±0.2	70.1±0.6	81.5±0.4
CPAT	89.8±0.7	88.0±0.4	88.9±0.5	89.8±0.7	80.6±0.6	85.0±0.6
CPC2	94.4±0.5	70.1±0.4	80.4±0.4	94.4±0.5	52.5±0.4	67.5±0.4
FEElnc	92.5±0.4	91.1±0.7	91.8±0.4	92.5±0.4	86.1±0.7	89.2±0.5
PLEK	98.1±0.2	70.2±0.7	81.8±0.4	98.1±0.2	55.1±0.3	70.5±0.2
longdist	98.6±0.2	51.4±0.2	67.6±0.2	98.6±0.2	55.1±0.2	70.7±0.2
lncRNA-MFDL	93.9±0.4	80.8±0.4	86.9±0.3	93.9±0.4	69.4±0.6	79.8±0.5
LncADeep	93.8±0.5	94.5±0.4	**94.2±0.3**	93.8±0.5	90.3±0.6	**92.0±0.5**

[a] The dataset is from GENCODE (lncRNAs and full- and partial-length mRNAs). LncADeep was trained for the model targeting transcripts including full- and partial-length.

[b] The composition of mRNAs in test set is 65% full-length and 35% partial-length, which matches the composition of full- and partial-length mRNAs in GENCODE dataset.

We thus compared the 10-fold cross-validation performances of LncADeep and other tools on transcripts including full- and partial-length. The results were listed in Table 2.8, where two test sets were used, one including both full-length and partial-length mRNAs whose composition (65 %full-length and 35% partial-length mRNAs) matches that of GENCODE dataset, and the other including only partial-length mRNAs (Figure 2.6). On the full- and partial-length test set, LncADeep achieves the highest harmonic mean of 94.2% with a noticeable advantage than others. Compared with the results for the full-

length test set, the performance of all tools on partial-length test set were seen brought down, clearly caused by the complication of partial-length transcripts. Among all the tools, LncADeep achieves the highest harmonic mean of 92.0% on partial-length test set. Although CPC has the highest sensitivity of 98.9% higher than LncADeep's 93.8%, CPC presents a much lower specificity of 57.5%. Meanwhile, the COME tool achieves the highest specificity of 96.1% higher than LncADeep's 90.3%, but it does a much lower sensitivity of 86.9%, leading to a harmonic mean of 91.2% lower than LncADeep's 92.0%. Despite achieving a comparable performance with LncADeep on partial-length test set, COME relies on experimental information and reference genome annotation, while LncADeep is an ab initio method which does not require these information and still outperforms COME. Besides, we plotted the mean ROC curves for lncRNA identification by LncADeep and other tools with 10-fold cross-validation on full- and partial-length human transcripts (Figures 2.8 and 2.9), and LncADeep achieved the highest AUC and outperformed other tools, which complied with the results in Table 2.8. Furthermore, we constructed various test sets by randomly combining full- and partial-length mRNAs (Figure 2.6) and compared the performance of LncADeep and other tools on them. Figure 2.10 illustrated five tools, LncADeep, COME, CPAT, FEElnc and lncScore with the best performances, while Figure 2.11 displayed the comparison for all tools. It is clear that the LncADeep tool outperforms all other tools consistently no matter how the ratio of partial-length mRNAs varies.

In summary, with a series of cross-validation tests both for full-length transcripts and partial-length transcripts, we show that the identification performance of LncADeep is higher than all other tools, even with the over-estimation for several of them trained by data overlapped with test sets.

2.3.2 Performance of lncRNA cross-species identification

In view of genetic conservation and diversity, it is worth expecting that the lncRNA predictors can be applied well in cross-species. To this end, we present the performance of

Table 2.9: Comparison of performances for cross-species lncRNA identification on full-length mouse transcripts.

Category	Tool	Full-length transcripts[a]		
		Sn (%)	Sp (%)	Hm (%)
Reference-based	COME	98.4	41.1	58.0
	lncScore	93.5	92.4	92.9
Reference-free	CPC	**98.6**	73.3	84.1
	CNCI	96.2	85.5	90.6
	CPAT	94.9	92.0	93.4
	CPC2	93.9	89.7	91.8
	FEElnc	94.8	92.7	93.7
	PLEK	90.2	75.5	82.2
	longdist	97.6	33.5	49.9
	lncRNA-MFDL	95.9	91.7	93.8
	LncADeep	97.0	**96.3**	**96.7**

[a] Full-length mouse dataset is composed of 12,529 lncRNAs (from GENCODE) and 29,739 mRNAs (from RefSeq).

lncRNA cross-species identification by LncADeep and other tools, with their algorithms trained on human data. Herein we chose the mouse for cross-species identification, as mouse is evolutionarily close to human, and moreover there are relatively abundant experimentally verified lncRNAs and mRNAs for mouse. As lncRScan-SVM cannot be used for cross-species identification, we did not include it. We first performed the test against the data for full-length transcripts, the results are shown in Table 2.9. LncADeep kept high accuracy (with harmonic mean of 96.7%) on lncRNA cross-species identification on full-length transcripts (Table 2.9). The harmonic mean of COME [102], CPC [69] and PLEK [71] dropped to less than 90%, while other tools remained above. Besides, we plotted the ROC curves as Figure 2.12, and LncADeep achieved the highest AUC of 99.61% and outperformed other tools, which conformed with the results in Table 2.9. Moreover, the specificity and harmonic mean of COME [102] dropped to less than 60%, which are rather lower than its performance on human dataset, indicating that COME might not be suitable for lncRNA cross-species identification. This might be COME is highly dependent on the experimental information and reference genome annotation, which might not

Table 2.10: Comparison of performances for cross-species lncRNA identification on full- and partial-length mouse transcripts.

Tool	Full- and partial-length transcripts[a]					
	Full- and partial-length[b]			Partial-length		
	Sn (%)	Sp (%)	Hm (%)	Sn (%)	Sp (%)	Hm (%)
COME	98.4	51.8	67.9	98.4	46.5	63.1
lncScore	93.5	90.9	92.1	93.5	83.6	88.2
CPC	**98.6**	70.0	81.9	**98.6**	56.0	71.5
CNCI	96.2	81.1	88.0	96.2	69.3	80.6
CPAT	89.9	90.4	90.2	89.9	82.0	85.8
CPC2	93.9	73.2	82.3	93.9	52.3	67.2
FEElnc	90.7	91.1	90.9	90.7	84.1	87.3
PLEK	90.2	66.3	76.4	90.2	50.5	64.8
longdist	97.6	47.9	64.2	97.6	54.8	70.2
lncRNA-MFDL	95.9	89.4	92.6	95.9	77.6	85.8
LncADeep	95.1	**93.3**	**94.2**	95.1	**87.7**	**91.2**

[a] Full- and partial-length mouse dataset is composed of 12,529 lncRNAs (from GENCODE) and 56,744 mRNAs (39,079 full-length and 17,665 partial-length, from GENCODE).

[b] The composition of mRNAs in test set is 70% full-length and 30% partial-length, which matches the composition of full- and partial-length mRNAs in GENCODE dataset.

be conservative across species.

In addition, we conducted cross-species lncRNA identification for mouse transcripts including full- and partial-length (Table 2.10 and Figures 2.13 and 2.14), LncADeep still achieved highest specificity (93.3%) and harmonic mean (94.2%). To comprehensively measure the performance of cross-species lncRNA identification on transcripts including full- and partial-length, we constructed various test sets by randomly combining full- and partial-length mouse mRNAs, where the percentage of full-length mRNAs ranges from 0% to 100% with a step of 5% (Figure 2.15). Figure 2.16 displayed the five tools with the best performance on cross-species lncRNA identification (Figure 2.17 plotted all benchmarked tools). Similar to Figure 2.10, LncADeep consistently outperformed other tools for cross-species lncRNA identification on mouse datasets. To summarize, the above results showed that LncADeep also outperformed state-of-the-art tools for cross-species lncRNA identification on mouse transcripts including full- and partial-length.

2.4 Discussion

For lncRNA identification, LncADeep has outperformed the state-of-the-art tools through integrating sequence and homology features with DBN. In particular, we designed two models, one targeting full-length transcripts and the other targeting transcripts including full- and partial-length.

To our knowledge, LncADeep is the first tool using the EDP of ORF for lncRNA identification. For the model targeting full-length transcripts, we used ORF length and coverage, the EDP of ORF, mean hexamer score, UTR coverage and GC content, Fickett nucleotide feature, and HMMER index, where the dimension of the EDP of ORF accounts for over two thirds. The success of LncADeep on lncRNA identification again supported the hypothesis that protein-coding and noncoding ORFs have different distributions in the EDP phase space, which can be caused by various selection pressure during evolution [117, 118]. Although most lncRNA identification tools can reach an accuracy over 90% on full-length transcripts, LncADeep has achieved the highest one with 97.7%. Actually, a tiny improvement of the accuracy matters: since there are a large number of lncRNAs (over 27,000 lncRNAs have been collected in GENCODE Release 24, and more are to be identified), one percent improvement on the accuracy indicates hundreds of correctly identified lncRNAs. With the development of sequencing technology, the third-generation sequencing technologies are emerging and being prevalent. The length of reads (ranges from 8,000 to 20,000 nt) sequenced by the third-generation sequencing technologies [125, 126] suffices for sequencing full-length transcripts, where LncADeep achieves high accuracy on lncRNA identification.

Current lncRNA identification is mainly based on transcripts assembled from RNA-seq data with the Next-generation sequencing technologies. Because of the length of short reads, assembled transcripts tend to be composed of full- and partial-length, while partial-length mRNAs can lead to partial CDS and complicate the identification of lncRNAs. For

partial-length mRNAs containing only partial CDS, the LORF-based model targeting on full-length transcripts can be error-prone. To address this problem, we modified the LORF-based model by searching the LCDS, which is chosen from two candidate sequences, ORF-based CDS and Hexamer-based CDS. For a partial-length mRNA, LCDS is more likely to overlap with the true CDS region. Accurately identifying the CDSs of mRNAs can help to avoid misclassifying mRNAs into lncRNAs. In this work, we use two strategies to detect putative CDS from transcripts, including the longest ORF (LORF) for full-length transcripts and the longest CDS (LCDS) for transcripts including full- and partial-length.

To evaluate the performance of these two strategies, we used LORF and LCDS to detect CDSs on full- and partial-length mRNAs, and compared the detected CDSs with known CDSs. The datasets were from GENCODE Release 24, including 58,528 full-length and 33,224 partial-length mRNAs of human, where the locations of CDSs are known. To quantify the performance of detecting CDS using the above two strategies, we used Mutual Overlap Rate (MOR) between detected CDS and known CDS, which is defined as follows:

$$MOR = \sqrt{\frac{L_{overlap}}{L_{known}} \times \frac{L_{overlap}}{L_{detected}}} \tag{2.8}$$

where $L_{overlap}$ refers to the length of the overlapped sequence between known CDS and detected CDS, L_{known} refers to the length of known CDS, and $L_{detected}$ represents the length of detected CDS. As $L_{overlap} \leq L_{known}$ and $L_{overlap} \leq L_{detected}$, we have $0 \leq MOR \leq 1$. The higher MOR, the better detected CDS.

Figures 2.18 and 2.19 illustrated the MOR on full- and partial-length mRNAs respectively. On full-length mRNAs (Figure 2.18), LCDS and LORF showed comparable performance. However, on partial-length mRNAs (Figure 2.19), LCDS outperformed LORF significantly. Therefore, when the transcripts are of full-length, it is better to use LORF strategy; when the transcripts are of full- and partial-length, it is more robust and much better to use LCDS strategy.

The reasons why LCDS performs better than LORF on partial-length transcripts might be, LORF is highly dependent on searching start and stop codons, but for partial-length transcript, the start and stop codons for true CDS might be missed, which can bias the locating of true CDS. As for LCDS, apart from locating LORF, it considers the Hexamer-based CDS, which will not be affected even if the start and stop codons are missed.

To evaluate the performance of various features used in the model for transcripts including full- and partial-length, we divided the features into three groups, i.e., LCDS-related features (including The EDP of the LCDS, the length and coverage of the LCDS, and Mean Hexamer Score, as these features are dependent on the LCDS), Fickett nucleotide feature, and HMMER index. Then, we conducted 10-fold cross-validation with these individual groups of features. Tables 2.11 and 2.12 displayed the performance of individual group of features in lncRNA identification. The datasets were human transcripts from GENCODE v24. When identifying lncRNAs from full-length transcripts, the LCDS-related features achieve the highest sensitivity, specificity, and harmonic mean, which is clearly higher than that of Fickett and HMMER features. For partial-length transcripts, LCDS-related features still showed highest performance with the average harmonic mean of 89.9%, which is 9.9% and 6.2% higher than that of Fickett and HMMER features, respectively. The performance comparison demonstrates the discriminative power of distinguishing lncRNAs from mRNAs using LCDS-related features.

Table 2.11: The performance of various features in lncRNA identification (only full-length mRNAs)

Features	Sn (%)	Sp (%)	Hm (%)
LCDS-related features	**96.3±0.3**	**87.9±0.7**	**91.9±0.4**
Fickett nucleotide features	79.7±0.8	60.3±0.7	68.7±0.6
HMMER index	95.4±0.3	84.5±0.6	89.6±0.3

In real dataset, the composition of full- and partial-length mRNAs is unknown, then it is not appropriate to train a specific classifier for any given real dataset. In this paper, we proposed to use majority voting, and results have shown the effectiveness of our

Table 2.12: The performance of various features in lncRNA identification (only partial-length mRNAs)

Features	Sn (%)	Sp (%)	Hm (%)
LCDS-related features	94.3±0.6	**85.9±0.5**	**89.9±0.4**
Fickett nucleotide features	84.7±0.9	75.8±0.9	80.0±0.5
HMMER index	95.4±0.3	74.5±0.6	83.7±0.4

strategy (Figures 2.10 and 2.16). Apart from using majority voting, we tested LncADeep on transcripts including various compositions of full- and partial-length mRNAs without majority voting, that is, we used all the negative training dataset (full- and partial-length mRNAs) for training. As illustrated in Figure 2.20, even if we do not use majority voting, LncADeep still showed overall better performance than other tools, indicating LncADeep's advantages. Although CNCI, lncScore and FEElnc also realized the partial-length mRNAs in real datasets, they did not address the various compositions of full- and partial-length mRNAs. To our knowledge, LncADeep is the first tool addressing this problem in real dataset and outperforms state-of-the-art tools.

As aforementioned, lncRNA identification methods can be divided into two groups, reference-based and reference-free methods. Reference-based tools which require the reference genome information (e.g, exon number) of transcript for lncRNA identification (e.g., lncScore [104], lncRScan-SVM [72]) might suffer some ambiguities for novel transcripts with alternative splicing structures. As for COME [102], which requires additional experimental information, it might not be suitable for cross-species lncRNA identification (Table 2.9). In contrast, LncADeep belongs to reference-free (or *ab initio*) methods, and one immediate advantage is: for non-model organisms without comprehensive genome annotation, LncADeep can still be used for lncRNA identification. Tables 2.9 and 2.10 have shown LncADeep's high performance on cross-species lncRNA identification. When conducting cross-species lncRNA identification, it is recommended that LncADeep be used for evolutionarily-close species. Currently, we have released models for human and mouse, and plan to train additional models for more model organisms.

To benchmark the time cost for lncRNA identification tools, we randomly selected 1000 transcripts, and tested tools on the same machine (3.6 GHz quad-core Intel i7-4790 processors, 24 GB memory, Ubuntu 16.04) and measured the elapsed (wall clock) time. We did not include reference-based tools, because these tools ignore the time cost for aligning transcripts to reference genome, which overestimates their efficiency. Benchmarked on 1,000 randomly chosen transcripts, the time cost of each tool was as following: CNCI (11 minutes), CPAT (2 seconds), CPC (1963 minutes), CPC2 (2 seconds), FEElnc (57 minutes), lncRNA-MFDL (143 minutes), longdist (5 seconds), PLEK (22 seconds), and LncADeep (9 minutes). CPAT, CPC2, longdist and PLEK were ranked as the fastest tools. This is reasonable as CPAT, CPC2, longdist and PLEK are alignment-free tools and only need to extract features by computing occurrences of *kmers*. Even though LncADeep requires aligning transcripts to Pfam database, LncADeep was as fast as CNCI, and faster than FEElnc, lncRNA-MFDL, and CPC. In a few words, the time cost of LncADeep was competitive among state-of-the-art tools.

2.5 Conclusions

With deep learning, we presented a novel *ab initio* lncRNA identification tool LncADeep based on deep belief network. Specifically, we designed models targeting full-length transcripts and transcripts including full- and partial-length. With the development of high-throughput sequencing technologies, the third-generation sequencing technologies, such as PacBio [125] and Nanopore [126], can generate full-length transcripts and are becoming prevalent. The accuracy of lncRNA identification of LncADeep reached up to 97.7% on full-length transcripts, which will be helpful for identifying novel lncRNAs from dataset generated by third-generation sequencing technologies. Meanwhile, current research on lncRNAs is mainly based on the Next-generation sequencing technologies, which tend to produce reads of short length, impeding the reconstruction of full-length transcripts, which in turn complicate the identification of lncRNAs. To address this problem, we pro-

posed LCDS-based model to describe partial-length transcripts. In addition, considering various compositions of full- and partial-length transcripts in real datasets, we integrated majority voting strategy. With a comprehensive performance comparison between our program LncADeep and other tools, results showed that LncADeep achieved better performance than state-of-the-art tools, including CPC, CPC2, CPAT, CNCI, COME, lncRScan-SVM, longdist, lncRNA-MFDL, lncScore, FEElnc and PLEK. Moreover, as a reference-free (or *ab initio*) tool, LncADeep does not require any reference genome annotation, and still outperformed other tools on cross-species lncRNA identification. It is expected that LncADeep can contribute on accurately identifying lncRNAs for RNA-seq community, and then facilitate the automatic genome annotation.

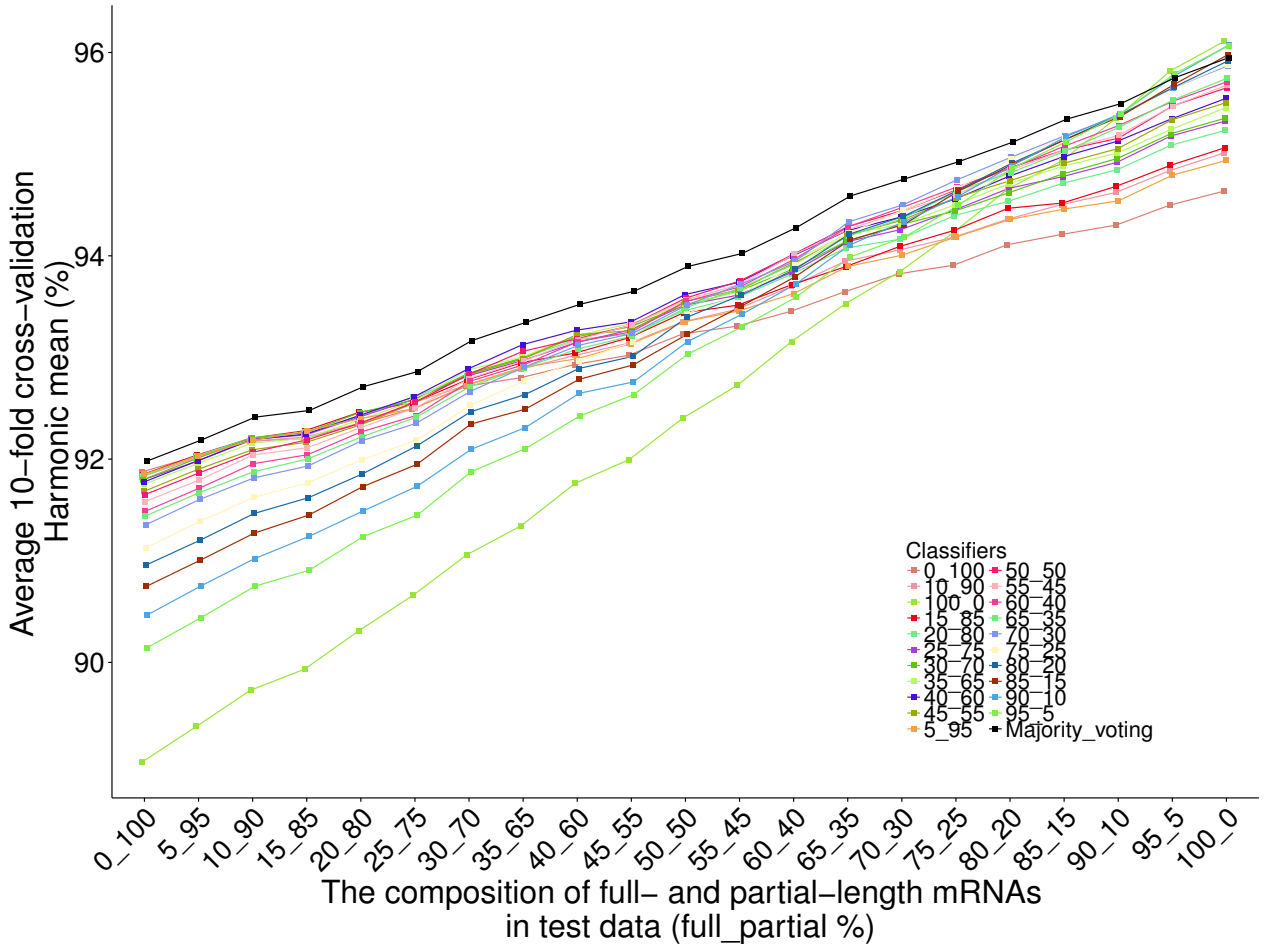


Figure 2.7: The lncRNA identification performance of majority voting and various classifiers on human transcripts.

To evaluate the lncRNA identification performance of majority voting and various classifiers on transcripts including various compositions of full- and partial-length mRNAs, we constructed various test datasets by randomly combining full- and partial-length mRNAs, where full- and partial-length mRNAs are from the remaining test dataset of every rotation (Figure 2.6), namely, the test dataset had no overlap with training dataset. The legend refers to the classifier trained on transcripts including a specific composition of full- and partial-length mRNAs. For example, Classifier 30_70 represents the classifier trained on transcripts including 30% full-length mRNAs and 70% partial-length mRNAs (the percentage is calculated within the mRNAs). As we can see, various classifiers show different performance on various test datasets, demonstrating that the composition of full- and partial-length mRNAs in training dataset can affect the lncRNA identification performance on the test dataset. In contrast, majority voting consistently outperformed other classifiers.

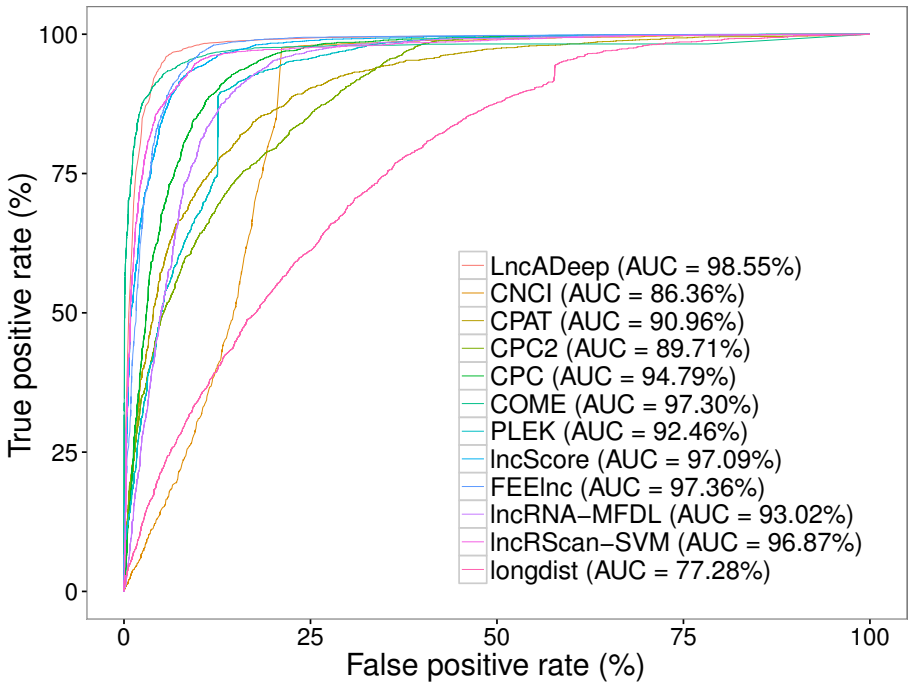


Figure 2.8: Mean ROC curves for lncRNA identification with 10-fold cross-validation on full- and partial-length human transcripts.

We plot the mean ROC curves for lncRNA identification by LncADeep and other tools with 10-fold cross-validation on full- and partial-length human transcripts, where the composition of mRNAs in test set is 65% full-length and 35% partial-length, which matches the composition of full- and partial-length mRNAs in GENCODE dataset. Herein, LncADeep achieves the highest AUC of 98.55% and outperforms other tools, which is consistent with the results in Table 2.8.

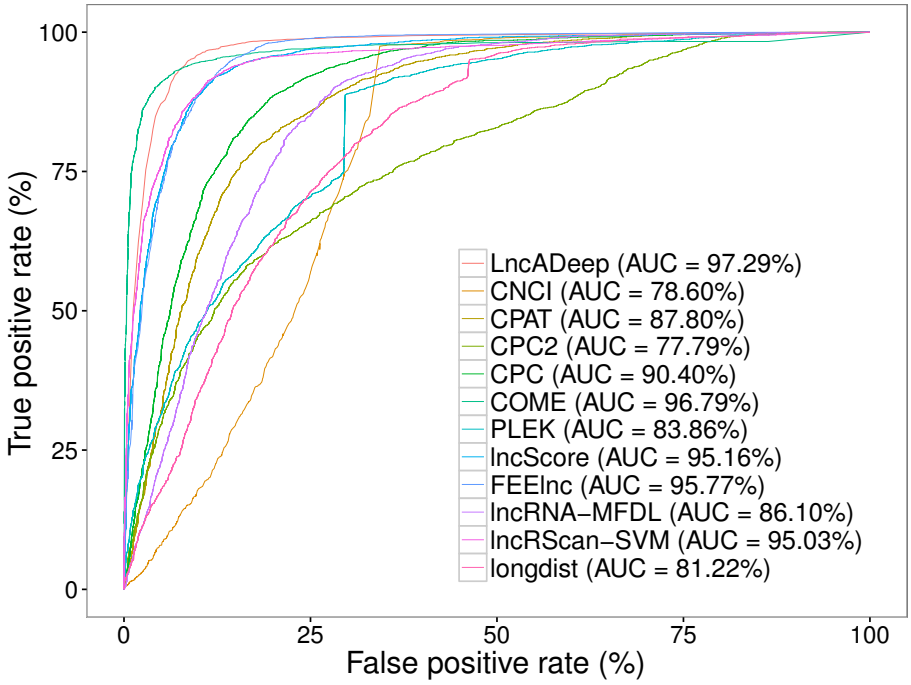


Figure 2.9: Mean ROC curves for lncRNA identification with 10-fold cross-validation on partial-length human transcripts.

We plot the mean ROC curves for lncRNA identification by LncADeep and other tools with 10-fold cross-validation on partial-length human transcripts, where the composition of mRNAs in test set is only partial-length. Herein, LncADeep achieves the highest AUC of 97.29% and outperforms other tools, which is consistent with the results in Table 2.8.

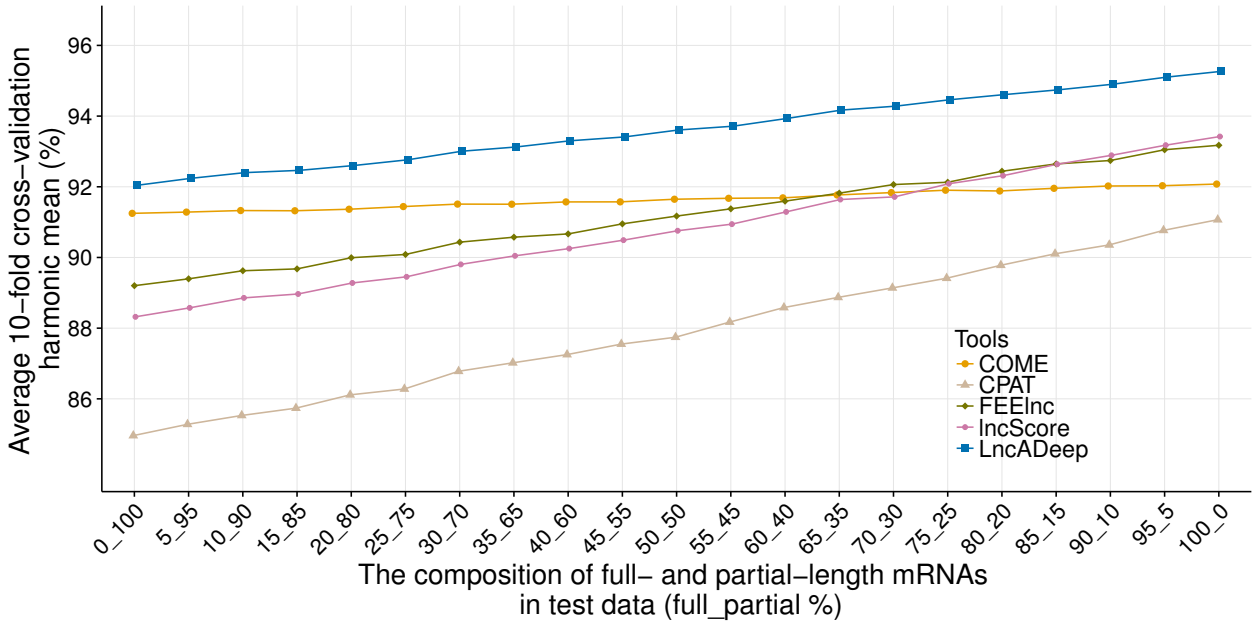


Figure 2.10: The performance of lncRNA identification on human transcripts with various compositions of full- and partial-length mRNAs.

LncADeep outperforms all other tools consistently no matter how the ratio of partial-length mRNAs varies. This figure illustrates five tools with with the best performances.

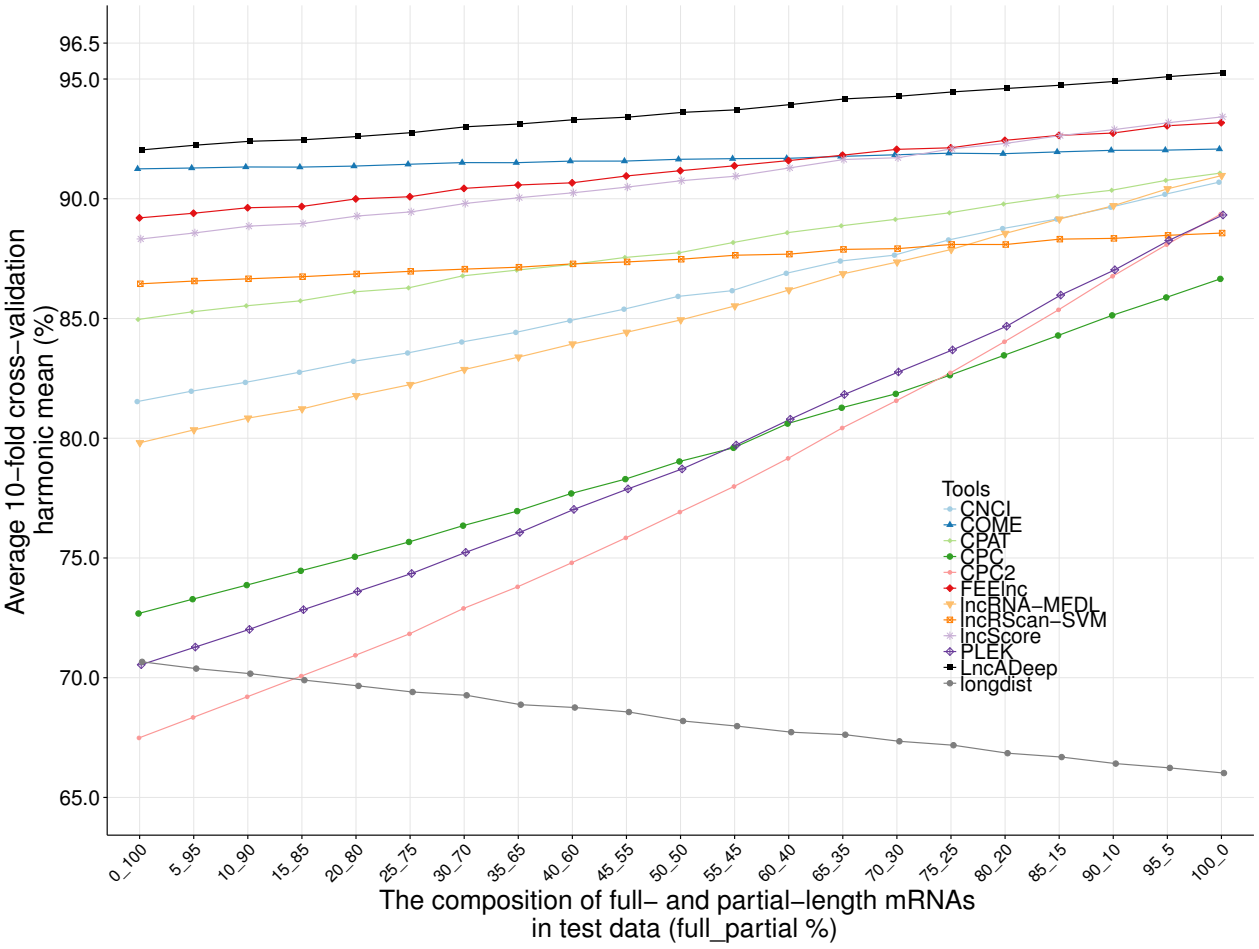


Figure 2.11: The performance of lncRNA identification on human transcripts with various compositions of full- and partial-length mRNAs.

LncADeep outperforms all other tools consistently no matter how the ratio of partial-length mRNAs varies.

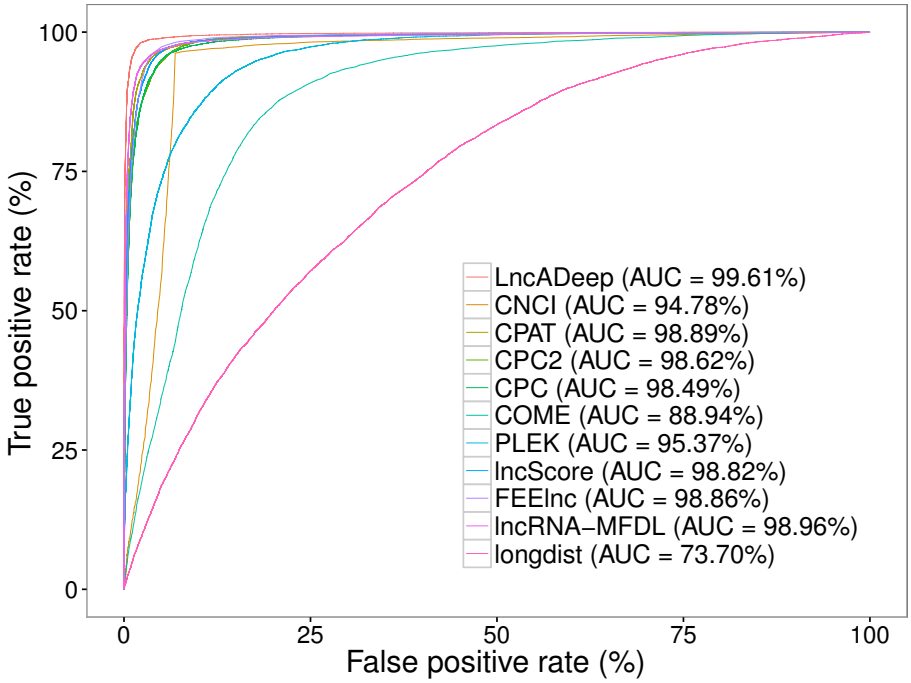


Figure 2.12: ROC curves for cross-species lncRNA identification on full-length mouse transcripts.

We plot ROC curves for cross-species lncRNA identification by LncADeep and other tools on full-length mouse transcripts. Herein, LncADeep achieves the highest AUC of 99.61% and outperforms other tools, which is consistent with the results in Table 2.9.

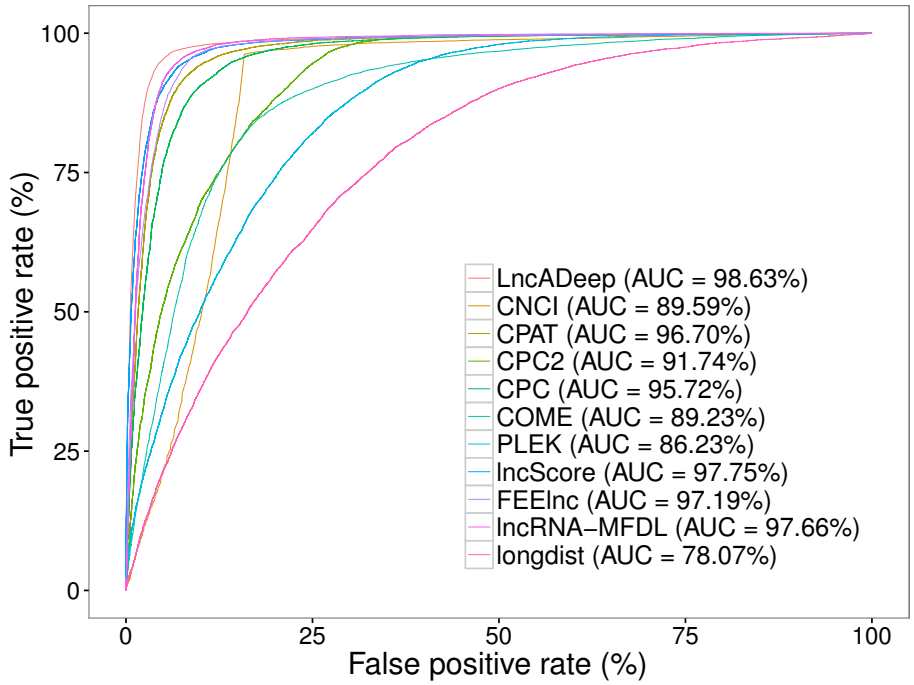


Figure 2.13: ROC curves for cross-species lncRNA identification on full- and partial-length mouse transcripts.

We plot ROC curves for cross-species lncRNA identification by LncADeep and other tools on full- and partial-length mouse transcripts, where the composition of mRNAs in test set is 70% full-length and 30% partial-length, which matches the composition of full- and partial-length mRNAs in GENCODE dataset. Herein, LncADeep achieves the highest AUC of 98.63% and outperforms other tools, which is consistent with the results in Table 2.10.

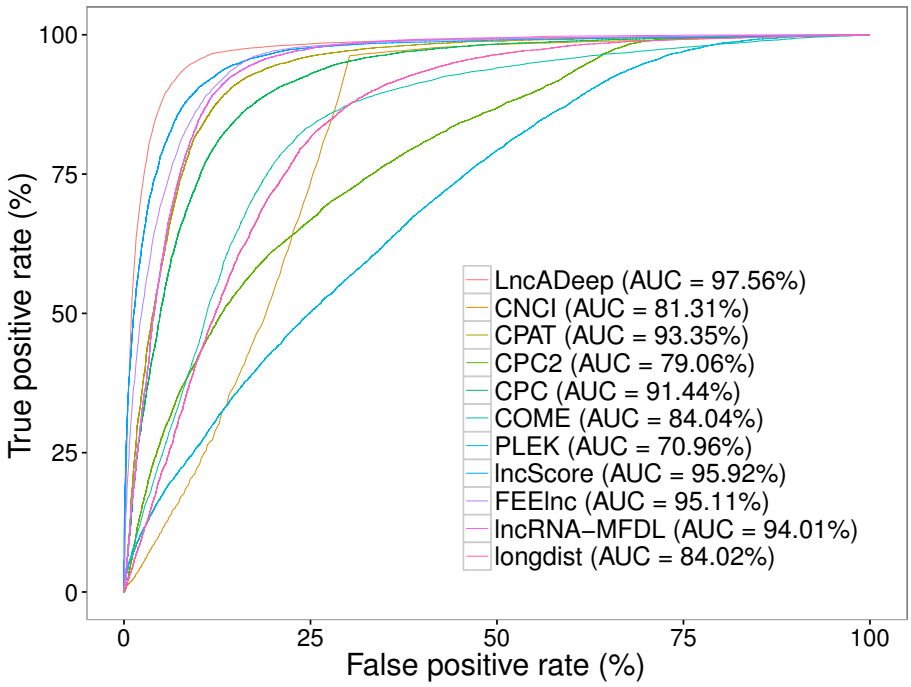


Figure 2.14: ROC curves for cross-species lncRNA identification on partial-length mouse transcripts.

We plot the ROC curves for cross-species lncRNA identification by LncADeep and other tools on partial-length mouse transcripts, where the composition of mRNAs in test set is only partial-length. Herein, LncADeep achieves the highest AUC of 97.56% and outperforms other tools, which is consistent with the results in Table 2.10.

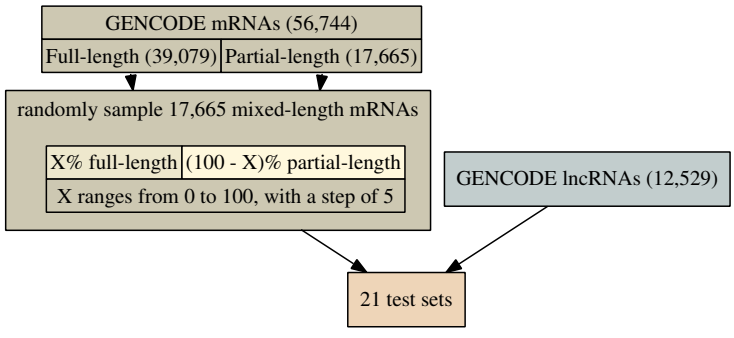


Figure 2.15: The construction of mouse test sets with various compositions of full- and partial-length mRNAs.

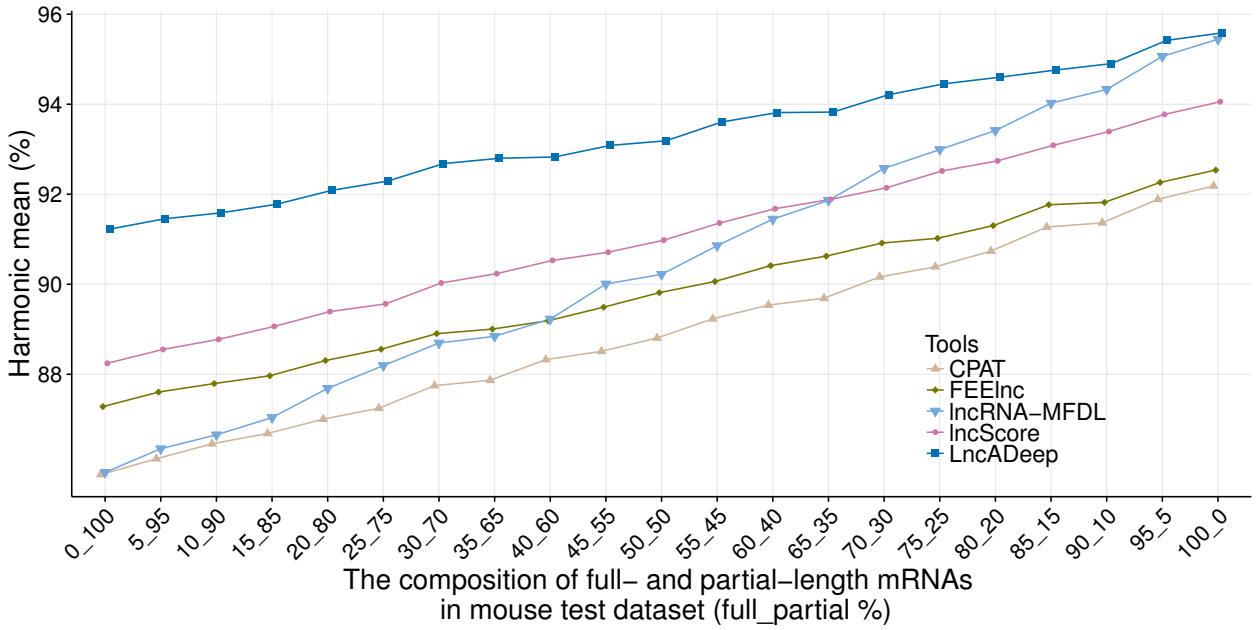


Figure 2.16: The performance of cross-species lncRNA identification on mouse transcripts with various compositions of full- and partial-length mRNAs.

LncADeep outperforms all other tools consistently for cross-species lncRNA identification on mouse datasets no matter how the ratio of partial-length mRNAs varies. This figure illustrates five tools with with the best performances.

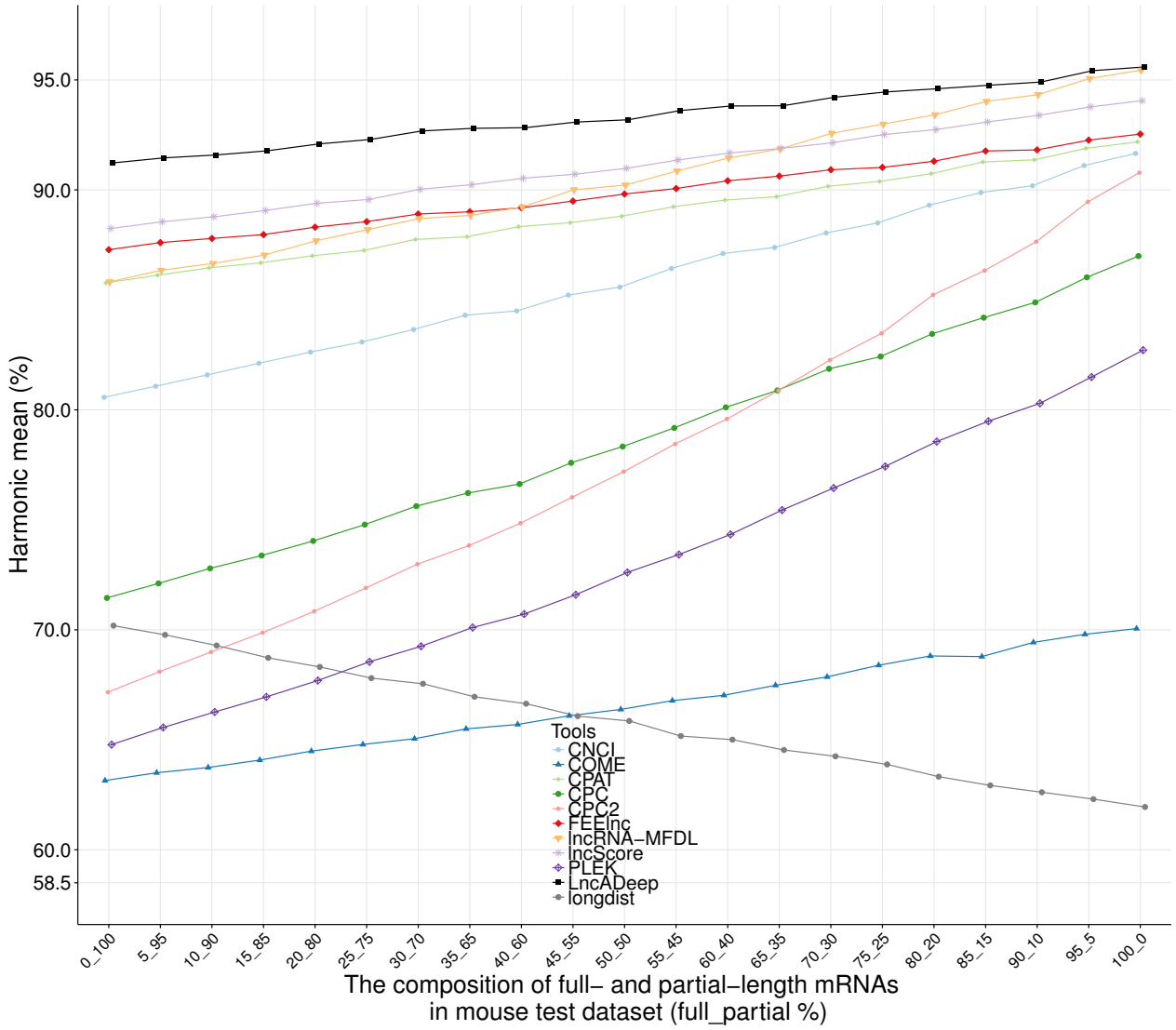


Figure 2.17: The performance of cross-species lncRNA identification on mouse transcripts with various compositions of full- and partial-length mRNAs.

LncADeep outperforms all other tools consistently for cross-species lncRNA identification on mouse datasets no matter how the ratio of partial-length mRNAs varies.

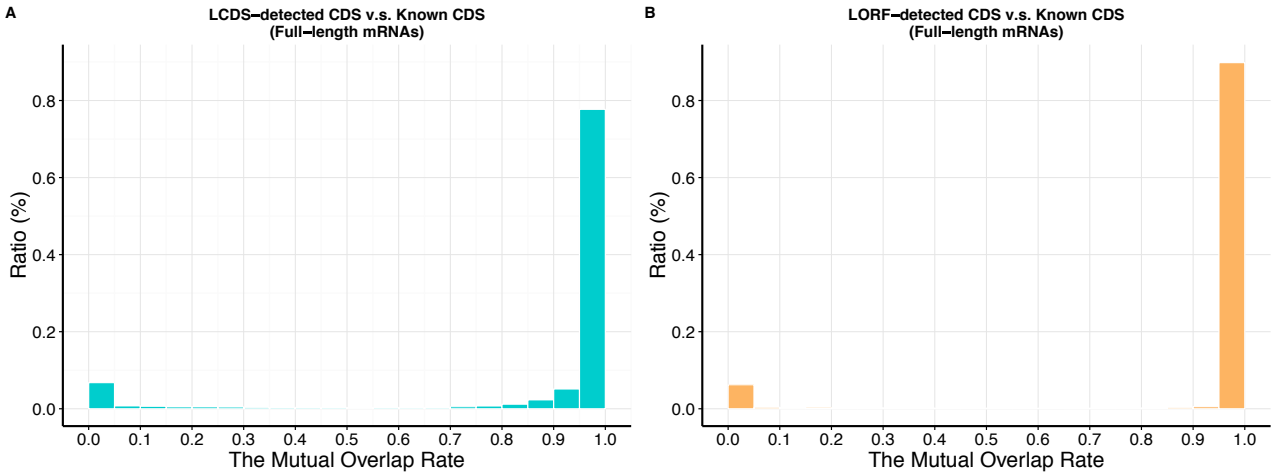


Figure 2.18: The Mutual Overlap Rate on full-length mRNAs.

Mutual Overlap Rate (MOR) quantify the performance of detecting CDS. On full-length mRNAs, LCDS and LORF showed comparable performance.

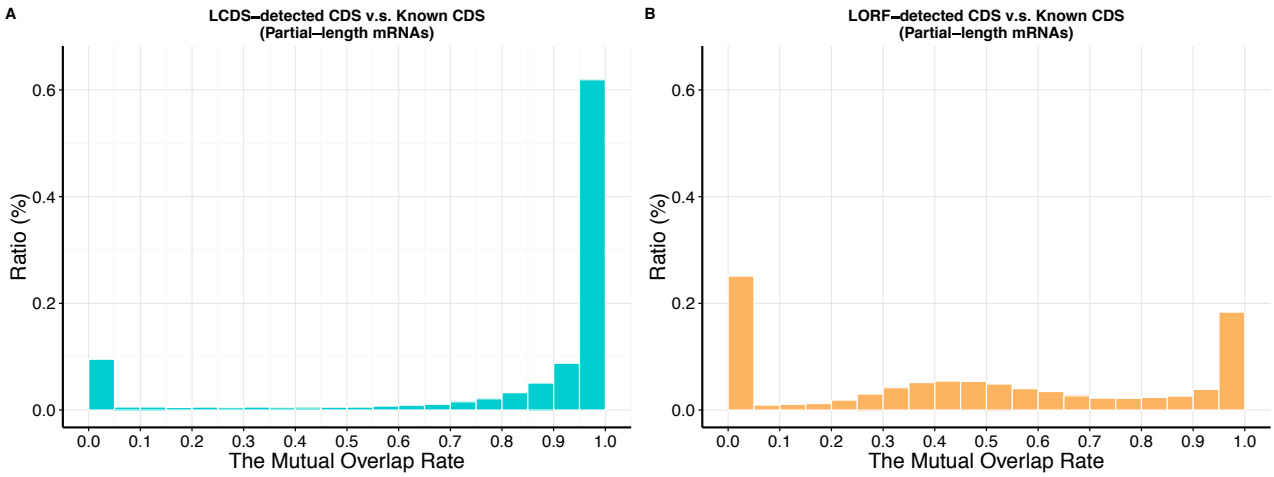


Figure 2.19: The Mutual Overlap Rate on partial-length mRNAs.

Mutual Overlap Rate (MOR) quantify the performance of detecting CDS. On partial-length mRNAs, LCDS outperformed LORF significantly.

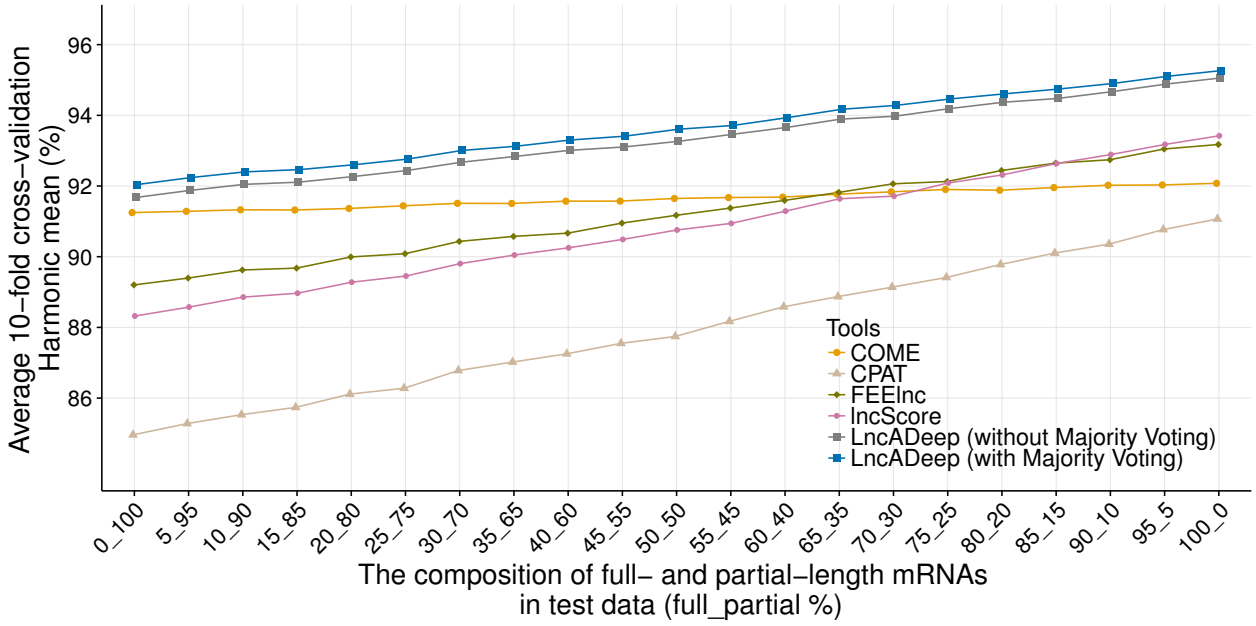


Figure 2.20: The lncRNA identification performance on human transcripts (with and without majority voting).

Apart from using majority voting, we tested LncADeep on transcripts including various compositions of full- and partial-length mRNAs without majority voting, that is, we used all the negative training dataset (full- and partial-length mRNAs) for training. As we can see in this figure, even without Majority Voting, LncADeep still showed overall better performance than other four tools, while LncADeep with majority voting showed highest performance. Here, only the five tools with the best performance were plotted.

CHAPTER 3
PREDICT LNCRNA-PROTEIN INTERACTIONS AND INFER LNCRNA FUNCTIONS

This chapter describes the method we proposed for predicting lncRNA-protein interactions based on a deep learning algorithm. The predicted interacting proteins are then used for inferring the functions of lncRNAs. The proposed method for predicting lncRNA-protein interaction and inferring lncRNA functions has been implemented into the tool named LncADeep (http://cqb.pku.edu.cn/ZhuLab/LncADeep).

3.1 Introduction

After identifying lncRNAs, the next step is to infer their functions. However, comprehensively characterizing lncRNAs is not simple since the functions of lncRNAs tend to be "diverse and complex" [85]. To exert biological functions, lncRNAs can interact with DNAs, RNAs, and proteins (lncRNA-protein interactions), and protein is confirmed to be the first and principle partner of lncRNAs [60, 50, 61]. Among the three categories of interactions, the interactions between lncRNAs and DNAs/RNAs are less studied, while lncRNA-protein interactions play the crucial roles in the functioning of lncRNA, providing satisfactory details of how lncRNAs exert functions in various biological processes [60]. For example, lncRNA HOTAIR can interact with polycomb repressive complex 2 (PRC2) and the CoREST-LSD1 complex to "specify the pattern of histone modification on target genes" [55]. Interacting with 81 proteins from chromatin modification, nuclear matrix and RNA remodeling pathways, Xist is an essential lncRNA for X chromosome inactivation [56]. Thus, identifying lncRNA-protein interactions will contribute to the understanding of lncRNA functions and mechanisms in biological process.

Although several experimental approaches, such as cross-linking and immunoprecipita-

Table 3.1: Tools for predicting lncRNA-protein interactions

Tool	Algorithm	Sequence features	Structure features	Year
catRAPID	Non-linear Discriminate	No	Yes	2011
GlobalScore	Non-linear Discriminate	No	Yes	2017
RPISeq	SVM and Random Forest	Yes	No	2011
lncPro	Linear Discriminate Method	No	Yes	2013
RPI-Pred	SVM	Yes	Yes	2015
rpiCool	Random Forest	Yes	No	2016
IPMiner	Deep neural network	Yes	No	2016

tion (CLIP) [127] and chromatin isolation by RNA purification (ChIRP) [78], are available for probing RNA-protein interactions, experimental approaches are expensive and time-consuming [106, 128]. In contrast, computational approaches are more convenient and rapid, and can employ experimentally verified datasets to infer lncRNA-protein interactions. Therefore, it is essential to develop computational approaches to predict lncRNA-protein interaction. Several methods are available (Table 3.1), such as catRAPID [129], GlobalScore [130], RPISeq [106], lncPro [86], RPI-Pred [107], rpiCool [108], and IPMiner [109], and these methods do not rely on the interaction networks among lncRNAs and proteins. They integrate sequence and/or structure features of RNAs and proteins to predict interactions (Table 3.1). catRAPID, GlobalScore and lncPro combine secondary structure, hydrogen-bonding and Van der Waals propensities; RPISeq, rpiCool, and IPMiner use *k-mer* based sequence information; RPI-Pred employs both *k-mer* based sequence features and predicted structure features of proteins and RNAs. To construct models, most of these tools use machine learning algorithms, such as SVM (RPISeq and RPI-pred), Random Forests (RPISeq and rpiCool), Linear Discriminate Method (lncPro) and stacked ensembling of neural network (IPMiner). A very recently developed tool, IPMiner, constructs a complicated model: first, it employs stacked autoencoder to extract hidden features from *k-mer* based sequence information; second, these features are fed to random forest models; and finally, stacked ensembling of neural networks are used to improve the prediction per-

formance [109]. However, these methods do not give functional annotations for lncRNAs, except outputting the interaction results.

LPIHN [131], [132], and LPBNI [133] are interaction network-based methods, which requires known information to construct interaction network for predicting potential interactions in the network. LPINH needs known protein-protein interactions, lncRNAs expression similarity, and known lncRNA-protein interactions to construct a heterogeneous network, and then infers lncRNA-protein interactions with random walk. Similarly, [132] constructs a heterogeneous lncRNA-protein network, but adopts a relevance search algorithm for interaction prediction. LPBNI requires known lncRNA-protein interactions to build a lncRNA-protein bipartite network, and then applies a propagation method to compute the interaction score. However, these network-based methods are highly dependent on known information and can predict only potential novel interactions in the network. For many lncRNAs whose lncRNA-protein interactions and lncRNA-lncRNA similarity are unknown, the interaction network cannot be constructed and the potential interactions cannot be inferred. Besides, the lacking of executive tools limits the application of LPBNI, [132], and LPIHN. Therefore, in this dissertation, we focus on network-independent methods.

For lncRNA functional annotation, we infer the functions of a lncRNA from its interacting proteins. Here, LncADeep adopts deep neural networks for predicting lncRNA-protein interaction using both sequence and structure information, achieving better performance compared with RPISeq, RPI-pred, rpiCool, IPMiner and lncPro. Then, LncADeep integrates KEGG [134] and Reactome [135] pathway enrichment analysis and functional module detection with the predicted interacting proteins. Thus, LncADeep provides the over-represented KEGG and Reactome pathways and functional modules as functional annotations for lncRNAs. As an *ab initio* lncRNA functional annotation tool, we expect that not only can LncADeep predict the interacting proteins of lncRNAs, but also provide helpful functional annotations for lncRNAs.

3.2 Materials and methods

3.2.1 Data description

To predict lncRNA-protein interactions, we obtained data from NPInter database [136], which collects experimentally verified lncRNA-protein interacting pairs. Since we focus on human lncRNA-protein interactions, we kept only the interacting pairs labeled with 'Homo sapiens' and 'ncRNA-protein binding', and removed the interacting pairs whose ncRNA is shorter than 200 nt. After removing redundant and obsolete interacting pairs (some lncRNAs were removed in the latest version of NONCODE, we filtered these lncRNA-related pairs), we finally obtained 6,204 lncRNA-protein interacting pairs consisting of 2,356 lncRNAs and 90 proteins. The sequences of lncRNAs and proteins were from NONCODE [137] and Uniprot [138] respectively. As NPInter provides only interacting lncRNA-protein pairs, which are considered as positive class, we need to generate non-interacting lncRNA-protein pairs as negative class. To construct a balanced negative dataset, we referred to the method used in [108, 106], i.e., pair lncRNAs and proteins, exclude all known interactions, and randomly keep 6,204 lncRNA-protein pairs as non-interacting ones.

3.2.2 Methods

To characterize lncRNA-protein interacting pairs, we use both sequence and structure information, which have been shown useful in [108, 106, 107]. The flowchart of predicting lncRNA-protein interaction is illustrated in Figure 3.1. For sequence information, each lncRNA is encoded with a 256-dimensional vector, which is the EDP of *4-mer* appearing in lncRNA sequence.

$$s_i = -\frac{1}{H} c_i \log c_i \tag{3.1}$$

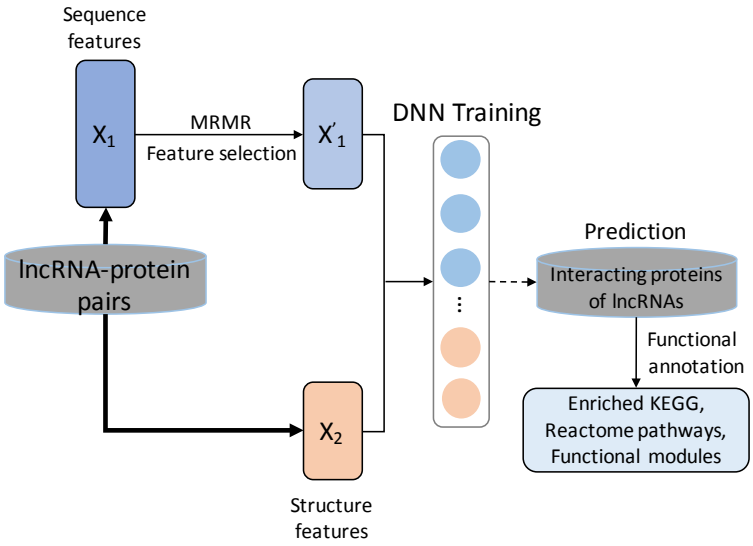


Figure 3.1: The flowchart of predicting lncRNA-protein interaction.

We use sequence and structure features for the prediction of lncRNA-protein interaction. X_1, $X_1^{'}$ and X_2 refer to sequence features, sequence features after feature selection, and structure features, respectively. The identified lncRNAs can be input for predicting lncRNA-protein interactions, then the interacting proteins can be used for inferring the functions of lncRNAs.

where $H = -\sum_{j=1}^{256} c_j \log c_j$ is the Shannon entropy, c_j is the abundance of *j*th *4-mer*, and $i, j = 1, \ldots, 256$ represents the index of the 256 *4-mers*.

Apart from the EDP of *4-mer*, we encode a lncRNA with the features used in lncRNA identification, including Fickett nucleotide feature and the features of LCDS (i.e., the EDP of the LCDS, LCDS length and coverage, and Mean hexamer score), which consists of a 47-dimensional vector.

Proteins are represented using conjoint triad method proposed in [139]. First, the 20 amino acids are divided into seven groups according to their dipole moments and side chain volume: {A, G, V}, {I, L, F, P}, {Y, M, T, S}, {H, N, Q, W}, {R, K}, {D, E}, and {C}. Then, each protein sequence is encoded with a 343-dimensional vector, which is the EDP of *3-mer* "in the 7-letter alphabet representation of the protein sequence [139]."

$$s_i = -\frac{1}{H} p_i \log p_i \tag{3.2}$$

where $H = -\sum_{j=1}^{343} p_j \log p_j$ is the Shannon entropy, p_j is the abundance of *j*th *3-mer*, and

$i, j = 1, \ldots, 343$ represents the index of the 343 *3-mers*.

In total, each lncRNA-protein pair is represented by a 646 ($4^4 + 7^3 + 47 = 646$) dimensional sequence feature vector. However, that many features can incur overfitting when the size of training data is relatively small, while the number of experimentally verified lncRNA-protein interacting pairs is limited (only 6204 interacting pairs are available). To alleviate overfitting, one approach is to use only the most discriminative features for model training [140]. Thus, we conduct feature selection with mRMR (minimal-redundancy-maximal-relevance criterion) [141]. As a feature selection method based on mutual information, mRMR can select a list of most characterizing features from candidate ones [141]. We tried selecting 100 features (the average 5-fold cross-validation harmonic mean was 90.7%) and 110 features (the average 5-fold cross-validation harmonic mean was 90.8%) using mRMR. Since the average 5-fold cross-validation accuracy of 110 features was higher, we chose to select 110 features. The detailed parameters for mRMR are list in Table 3.2.

For structure information, we use the structure features of lncRNA and protein, including secondary structure, hydrogen-bonding and Van der Waals propensities [86]. To predict the secondary structure of a lncRNA, we use RNAfold [113]. However, RNAfold can be very time consuming as its time complexity is $O(L^3)$, where L is the length of sequence. To address this problem, we use a naive segmentation method to segment lncRNA sequence before predicting its secondary structure, since sequence segmentation can enhance the prediction accuracy and decrease the time cost [142]. To choose a segment length for

Table 3.2: The parameters of mRMR

Options	Value
-t [threshold]	0.2
-n [number of features]	110
-s [MAX number of samples]	12408

[a] For the other options not list here, we used its default parameters.

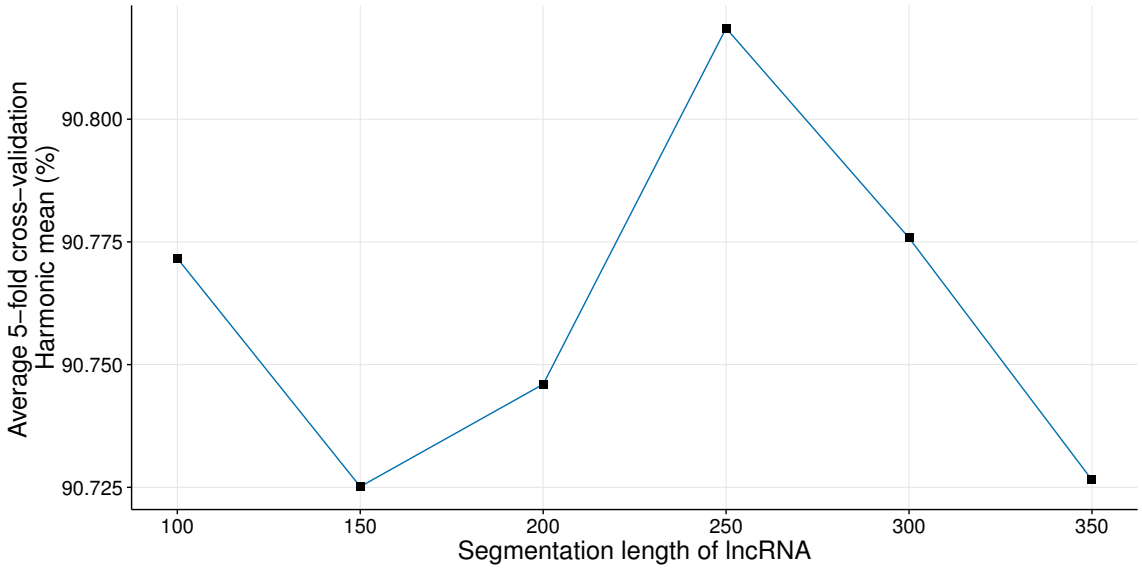


Figure 3.2: The average 5-fold cross-validation accuracy of predicting lncRNA-protein interaction.

To choose a segment length for lncRNA, we evaluated the performance of various segment lengths (ranging from 100 nt to 350 nt with a step of 50 nt) on predicting lncRNA-protein interaction. This figure displayed the average 5-fold cross-validation harmonic mean of various segment lengths. Considering the balance between the time cost (please refer to https://www.tbi.univie.ac.at/RNA/) and the accuracy, we chose n = 250 to segment lncRNA for secondary structure prediction.

lncRNA, we evaluated the performance of various segment lengths (ranging from 100 nt to 350 nt with a step of 50 nt) on predicting lncRNA-protein interaction. Figure 3.2 displayed the average 5-fold cross-validation accuracy of various segment lengths. Considering the balance between the time cost (please refer to https://www.tbi.univie.ac.at/RNA/) and the accuracy, we chose n = 250 to segment lncRNA for secondary structure prediction. First, a lncRNA is segmented into fragments with at most $n = 250nt$; then RNAfold is used to predict the secondary structure for each fragment; in the end, the secondary structure of each fragment are concatenated as the secondary structure of the lncRNA.

For the other structure features, we follow the process mentioned in lncPro [86]. The structure information of a lncRNA-protein pair is encoded using a 80-dimensional vector (Table 3.3). The sequence and structure features of lncRNA-protein pairs are combined for predicting lncRNA-protein interaction.

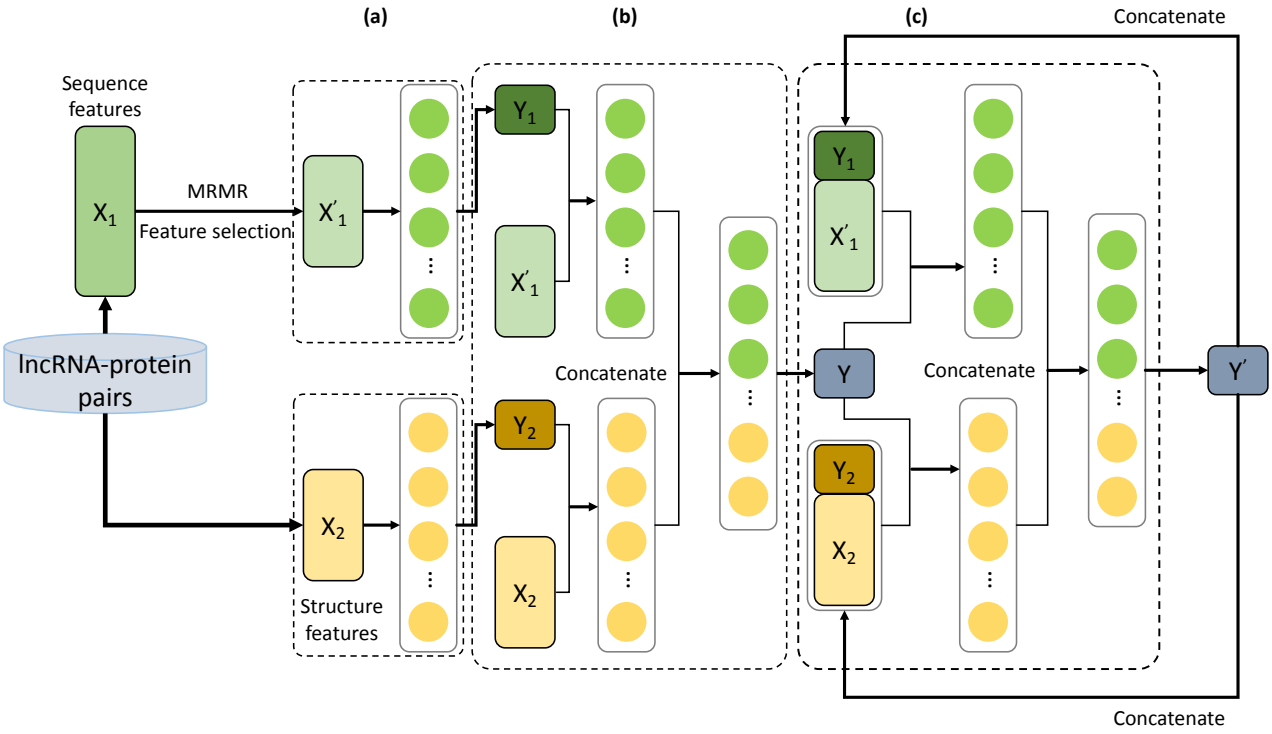


Figure 3.3: The structure of DNN for predicting lncRNA-protein interaction.

(a) The first step is to train two neural networks using only sequence or structure features separately and then output the prediction results. **(b)** The second step is to employ the prediction results from (a) as additional features and concatenate them to sequence and structure features, and use both sequence and structure features for training. **(c)** The third step is to use the prediction results from (b) as additional features and concatenate them again to sequence and structure features. Then another neural network will be trained with the features and output the prediction results. The prediction results can then be concatenated to the features again for training which leads to a stacking neural network.

3.2.3 Deep neural network

We construct a deep neural network (DNN) for predicting lncRNA-protein interaction. Inspired by the deep stacking network proposed in [143], the DNN is built as follows (Figure 3.3). First, we train two neural networks using sequence or structure information separately, where we can obtain the prediction results from sequence and structure

Table 3.3: The dimensions of the structure feature vector

Type	Secondary structure	Hydrogen-bonding	Van der Waals
lncRNA	10-dimension	10-dimension	10-dimension
protein	10-dimension	20-dimension	20-dimension

Table 3.4: The neural network architecture for sequence features

Layers	Nodes
Input layer	Sequence features (110 dim)
1st hidden layer	Dense(30, input_dim=110)
2nd hidden layer	Dense(20, activation='softmax')
Output layer:	Dense(1, activation='sigmoid')

Table 3.5: The neural network architecture for structure features

Layers	Nodes		
Input layer	Secondary Sequence features (20 dim)	Hydrogen-bonding features (30 dim)	Van der Waals features (30 dim)
1st hidden layer	Dense(20, input_dim=20)	Dense(30, input_dim=30)	Dense(30, input_dim=30)
	Concatenate		
Merged layer	80 dim		
1st hidden layer	Dense(20, input_dim=80, activation='softmax')		
2nd hidden layer	Dense(10, activation='softmax')		
Output layer	Dense(1, activation='sigmoid')		

information (Tables 3.4 and 3.5). Second, the prediction results from the first step are concatenated with the original sequence and structure features (as new features) to train another neural network, where we can get prediction results using both sequence and structure information (Table 3.6). Third, the prediction results from the second step are again concatenated with sequence and structure features to train another neural network, which can be iterated several times (Table 3.7) The rationale is that the prediction results from the previous step can offer additional helpful information. We use Keras 1.2.2 library (https://github.com/fchollet/keras/) to implement the DNN.

3.2.4 Infer the functions of lncRNA

As NPInter provides only interacting lncRNA-protein pairs, we need to generate non-interacting lncRNA-protein pairs as negative class. In this study, we follow the approach used in several methods, such as RPISeq [106], rpiCool [108], and RPI-pred [107], i.e., pairing 2,356 lncRNAs and 90 proteins and removing known interacting lncRNA-protein

Table 3.6: The neural network architecture for the second step

Layers	**Nodes**	
Input layer	Sequence features (110 + 1 dim)	Sturcture features (80 + 1 dim)
1st hidden layer	Dense(30, input_dim=111)	Dense(30, input_dim=81)
2nd hidden layer	Dense(20, activation='softmax')	Dense(20, activation='softmax')
	Concatenate	
Merged layer	40 dim	
1st hidden layer	Dense(20, activation='softmax'), Dropout(0.5)	
2nd hidden layer	Dense(20, activation='softmax'), Dropout(0.5)	
3rd hidden layer	Dense(10, activation='softmax'), Dropout(0.5)	
Output layer	Dense(1, activation='sigmoid')	

Table 3.7: The neural network architecture for the third step

Layers	**Nodes**	
Input layer	Sequence features (111 + i dim)	Sturcture features (81 + i dim)
1st hidden layer	Dense(30, input_dim=111+i)	Dense(30, input_dim=81+i)
2nd hidden layer	Dense(20, activation='softmax')	Dense(20, activation='softmax')
	Concatenate	
Merged layer	40 dim	
1st hidden layer	Dense(20, activation='softmax'), Dropout(0.5)	
2nd hidden layer	Dense(20, activation='softmax'), Dropout(0.5)	
3rd hidden layer	Dense(10, activation='softmax'), Dropout(0.5)	
Output layer	Dense(1, activation='sigmoid')	

pairs, and then we obtained 205,836 non-interacting pairs. However, 6,204 interacting pairs and 205,836 non-interacting pairs led to a class-imbalance. To handle class-imbalance, under-sampling the majority class is an efficient strategy, which is what we used for benchmarking the lncRNA-protein interaction tools in the main text. However, when using the trained model for predicting, it might suffer some drawbacks such as losing potential useful data [144]. In the release version of LncADeep, to make full use of the non-interacting pairs for predicting lncRNA-protein interactions, we use EasyEnsemble [144]. As a straightforward strategy to address the class-imbalance problem, EasyEnsemble [144] first indepen-

dently samples several subsets from the majority class, then for each subset, a classifier will be trained, and finally all classifiers will be combined for the final classification. In this study, as the number of non-interacting pairs is around 33 times of that of interacting pairs, first of all, we sampled 33 subsets from the non-interacting pairs. Second, for each subset, we trained a classifier with the positive class (interacting pairs), and thus we obtain 33 classifiers and combine the results from 33 classifiers using majority voting. Finally, the predicted interacting proteins are used for inferring the functions of lncRNAs.

To annotate the functions of a lncRNA from its predicted interacting proteins, LncADeep integrates KEGG and Reactome pathway enrichment analysis and functional module detection. Enrichment analysis is a prevalent approach to identify some over-represented biological functions that shared by a group of genes or proteins based on statistical analysis, which can be referred to in [145]. To conduct KEGG and Reactome pathway enrichment analyis, first, we downloaded reviewed human protein sequences from Uniprot database [138], and finally obtained 20,121 protein sequences after filtering. Herein, we filtered protein sequences whose length is shorter than 15 AA and longer than 15,000 AA, as Predator [146] was used to predict the secondary structure of proteins and could not process sequence whose length are not in that range. Then, LncADeep predicts the interacting proteins of lncRNAs from the 20,121 proteins and conducts the function annotations with the predicted interacting proteins. For KEGG and Reactome pathway enrichment analysis, LncADeep uses Fisher's exact test for the significance test, Benjamini-Hochberg (BH) method [147] for the multiple testing correction, and keeps enriched pathways whose adjusted p-value is less than 0.05. Proteins usually function as modules, where protein-protein interaction (PPI) information can be used for detecting functional modules [148], and interpreting the functional modules derived from the interacting proteins of lncRNAs can offer some helpful information for the functions of lncRNAs. To detect functional modules in the interacting proteins of a lncRNA, LncADeep uses Markov Clustering (MCL) [149] by integrating PPI information provided by HIPPIE database [150], which collects experimen-

tally verified interactions from various reliable sources. To display the functional modules, we adopt iGraph (http://igraph.org/r/) [151] to plot their interacting networks.

3.2.5 Evaluation metrics

To evaluate the prediction performance of the LncADeep and other existing tools, we use two independent quantities, Sn (sensitivity) and Sp (specificity), and a balanced quantity Hm (harmonic mean of sensitivity and specificity), they are defined as follows:

$$\begin{aligned} \text{Sn} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{Sp} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Hm} &= \frac{2 \times \text{Sn} \times \text{Sp}}{\text{Sn} + \text{Sp}} \end{aligned} \tag{3.3}$$

where TP, TN, FP, and FN represents true positive, true negative, false positive, and false negative respectively. Herein Sn measures the ratio of actual positives that are correctly identified, Sp measures the ratio of true positives in all predicted positives, and Hm is a composite measure of sensitivity and specificity [118]. For the prediction of lncRNA-protein interaction, interacting lncRNA-protein pairs are regarded as positive class, and non-interacting ones are as the negative.

3.3 Results

3.3.1 Performance of lncRNA-protein interaction prediction

To benchmark lncRNA-protein interaction prediction performance, we compared LncADeep with lncPro [86], RPISeq [106], RPI-pred [107], rpiCool [108] and IPMiner [109], which are state-of-the-art tools for predicting RNA-protein interactions, using 5-fold cross validation. Among these tools, IPMiner was the most recently developed and provides retraining option. Thus, we retrained IPMiner, but for the others, we used the tools (lncPro and rpiCool) or online servers (RPISeq and RPI-pred) as provided. Considering the unavailability

of stand-alone tools of catRAPID [129] and GlobalScore [130], we tried to submit lncRNA-protein pairs online and download the results, however, it was time-consuming and could not be conducted in large scale. Therefore, we did not include catRAPID and GlobalScore for benchmarking.

The harmonica mean of LncADeep reached up to 90.8%, outperforming the other five tools (Table 3.8). Besides, we noted that the specificity and the harmonic mean of lncPro [86], RPISeq [106] and RPI-pred [107] was much lower than that of rpiCool [108], IPMiner [109], and LncADeep, suggesting that lncPro, RPISeq and RPI-pred tend to predict most lncRNA-protein pairs as interacting ones. This might be the pre-built models of lncPro, RPISeq, and RPI-pred were trained on datasets containing RNA-protein interacting pairs, where the length of RNAs were much shorter than that of lncRNAs, thus limiting the application of these methods on predicting lncRNA-protein interaction. Nevertheless, LncADeep achieved better performance on the prediction of lncRNA-protein interaction when compared with rpiCool and IPMiner.

3.3.2 Performance of lncRNA functional annotation

Accurate lncRNA-protein interactions can help to infer the functions of lncRNAs, we thus used our tool LncADeep to annotate the functions for the 27,384 lncRNAs collected in GENCODE Release 24. The functional annotations including KEGG and Reactome path-

Table 3.8: Comparison of performances for predicting lncRNA-protein interaction with 5-fold cross-validation.

Tools	Sn(%)	Sp(%)	Hm(%)
lncPro	80.3±0.9	52.2±0.6	63.2±0.4
RPISeq (RF)	**99.1±0.2**	50.1±0.1	66.5±0.1
RPISeq (SVM)	93.5±0.7	50.2±0.2	65.3±0.4
RPI-pred	88.0±0.3	49.8±0.6	63.6±0.5
rpiCool	92.0±0.8	83.3±0.8	87.5±0.6
IPMiner	89.8±1.1	**85.6±0.7**	87.6±0.6
LncADeep	97.0±0.5	85.4±0.8	**90.8±0.4**

[a] RPISeq has two prediction modes, including SVM and random forest (RF).

KEGG_path_ID	KEGG_pathway	p_value	adj_p_value
PATH:ko05224	Breast cancer	0.00031	0.00799
PATH:ko04916	Melanogenesis	0.00099	0.00921
PATH:ko04390	Hippo signaling pathway	0.00109	0.00921
PATH:ko05217	Basal cell carcinoma	0.00142	0.00921
PATH:ko04610	Complement and coagulation cascades	0.00243	0.01261
PATH:ko04016	MAPK signaling pathway - plant	0.0068	0.02211
PATH:ko04512	ECM-receptor interaction	0.00522	0.02211
PATH:ko00910	Nitrogen metabolism	0.00638	0.02211
PATH:ko05150	Staphylococcus aureus infection	0.00951	0.02748
PATH:ko01210	2-Oxocarboxylic acid metabolism	0.01536	0.03995
PATH:ko01230	Biosynthesis of amino acids	0.01851	0.04293
PATH:ko04976	Bile secretion	0.02477	0.04293
PATH:ko00270	Cysteine and methionine metabolism	0.02047	0.04293
PATH:ko04310	Wnt signaling pathway	0.02201	0.04293
PATH:ko04066	HIF-1 signaling pathway	0.02466	0.04293
PATH:ko05014	Amyotrophic lateral sclerosis (ALS)	0.02693	0.04376
PATH:ko04011	MAPK signaling pathway - yeast	0.02911	0.04452
PATH:ko00190	Oxidative phosphorylation	0.03092	0.04466
PATH:ko04970	Salivary secretion	0.04037	0.04664
PATH:ko04971	Gastric acid secretion	0.04168	0.04664
PATH:ko04672	Intestinal immune network for IgA production	0.04267	0.04664
PATH:ko04150	mTOR signaling pathway	0.04306	0.04664
PATH:ko04260	Cardiac muscle contraction	0.03685	0.04664
PATH:ko04340	Hedgehog signaling pathway	0.03606	0.04664
PATH:ko04392	Hippo signaling pathway - multiple species	0.04878	0.04878
PATH:ko04933	AGE-RAGE signaling pathway in diabetic complications	0.04818	0.04878

Figure 3.4: An example of over-represented KEGG pathways.

This is an example of over-represented KEGG pathways for one alternative transcript of the lncRNA HOTAIR.

ways and functional modules are provided as supplementary files that can be downloaded online. Here, we focused on KEGG and Reactome pathways (Figure 3.4), which are more explicit for annotating functions than functional modules (Figure 3.5). Consequently, LncADeep annotated 702,675 associations with 140 KEGG pathways and 1,839,272 associations with 422 Reactome pathways for the 27,384 lncRNAs, with an average of 25 KEGG and 67 Reactome pathways associated with each lncRNA, conforming with the complexity of lncRNA functions [60, 50, 61]. To our best knowledge, LncADeep is the first tool that can automatically predict lncRNA-protein interactions and provide rich inferred functional annotations for lncRNAs.

Since IPMiner ranked next to LncADeep on lncRNA-protein interaction prediction, as a comparison, we used IPMiner [109] (its release version) to predict interacting proteins for 100 randomly sampled lncRNAs (IPMiner is very time-consuming and we cannot used it for all the 27,384 lncRNAs, and thus we randomly sampled 100 lncRNAs) and then conducted KEGG and Reactome pathway enrichment analysis with the predicted interacting proteins. For these 100 randomly sampled lncRNAs, LncADeep annotated 2,555 associations with 78 KEGG pathways and 6,697 associations with 216 Reactome pathways, with an average of 25 KEGG and 67 Reactome pathways associated with each lncRNA, matching the previous results. In contrast, using IPMiner-predicted interacting proteins, we only obtained 473 associations with 37 KEGG pathways and 2,840 associations with 84 Reactome pathways, with an average of only 5 KEGG and 28 Reactome pathways associated with each lncRNA, which are much smaller than that of LncADeep, indicating that LncADeep can give more detailed functional annotations for lncRNAs. In fact, as LncADeep outperformed IPMiner on predicting lncRNA-protein interactions, LncADeep is expected to provide better functional annotations. For lack of a gold standard dataset for lncRNA functions, it is difficult to quantitatively evaluate the performance of inferred functions of lncRNAs. Therefore, to demonstrate that LncADeep can provide helpful suggestions on lncRNA functioning, we took four well-studied lncRNAs as examples through comparing their inferred functions (by LncADeep and IPMiner) with the reported functions from literatures.

3.3.3 Examples for lncRNA functional annotation

HOTAIR.

As an oncogenic factor and a potential biomarker for various cancers, HOTAIR has been associated with several cancers [152], such as breast cancer [51] and cervical cancer [153], and LncADeep's annotations for HOTAIR comply with these known functions. First, LncADeep associates HOTAIR with KEGG pathways such as PATH:ko05224 Breast can-

cer, PATH:ko05200 Pathways in cancer, PATH:ko04330 Notch signaling pathway [153], and PATH:ko04310 Wnt signaling pathway [154]. Second, LncADeep-predicted Reactome pathways show that HOTAIR might be involved with R-HSA-1257604 PIP3 activates AKT signaling [155], R-HSA-216083 Integrin cell surface interactions, and R-HSA-5628897 TP53 Regulates Metabolic Genes, which are related to cell adhesion, tissue and organ morphogenesis, and cancers. LncADeep still outputs other enriched Reactome pathways. In fact, the over-represented KEGG and Reactome pathways complement each other. In contrast, using IPMiner-predicted interacting proteins, only two over-represented KEGG pathways including PATH:ko05200 Pathways in cancer and PATH:ko04014 Ras signaling pathway were detected, which cannot give detailed annotations for HOTAIR.

ANRIL.

ANRIL is involved in various human diseases, such as Melanoma [156], breast cancer [157], basal cell carcinoma [158], and cardiovascular disease [159]. LncADeep-predicted KEGG pathways for ANRIL conform with its known functions, namely PATH:ko05218 Melanoma, PATH:ko05224 Breast cancer, PATH:ko05217 Basal cell carcinoma, PATH:ko05200 Pathways in cancer, PATH:ko04260 Cardiac muscle contraction and PATH:ko04310 Wnt signaling pathway [160]. As a contrast, using the predicted interacting proteins by IPMiner for ANRIL, we obtained only three over-represented KEGG pathways, i.e., PATH:ko05200 Pathways in cancer, PATH:ko04015 Rap1 signaling pathway and PATH:ko04151 PI3K-Akt signaling pathway. Compared with IPMiner, LncADeep can not only offer predicted interacting proteins, but also give detailed annotation of ANRIL. Apart from the above diseases, LncADeep still shows that ANRIL can be involved with some other diseases such as PATH:ko05310 Asthma, PATH:ko05014 Amyotrophic lateral sclerosis, and PATH:ko05166 HTLV-I infection etc., which need further validation.

SPRY4-IT1.

Derived from an intron of the SPRY4 gene situated at chromosome 5q31, SPRY4-IT1 has an important role in several tumors, such as melanoma [161] and breast cancer [162]. LncADeep predicts that SPRY4-IT1 can be involved in PATH:ko05224 Breast cancer, PATH:ko04151 PI3K-Akt signaling pathway, PATH:ko05217 Basal cell carcinoma [163] and PATH:ko04310 Wnt signaling pathway [164], which were also supported by the above known functions. In contrast to LncADeep, IPMiner-predicted proteins can only give one over-represented KEGG pathway (PATH:ko05200 Pathways in cancer).

TUNAR.

Located at chromosome 14q32, TUNAR lncRNA has been linked to human neurodegenerative diseases [165], diabetes [166], and breast cancer [167]. LncADeep shows that TUNAR is associated with PATH:ko05010 Alzheimer's disease, PATH:ko05012 Parkinson's disease, PATH:ko05014 Amyotrophic lateral sclerosis (ALS), PATH:ko05224 Breast cancer, PATH:ko05200 Pathways in cancer and PATH:ko04933 AGE-RAGE signaling pathway in diabetic complications, which comply with its known functions. Besides, PATH:ko04151 PI3K-Akt signaling pathway, PATH:ko04510 Focal adhesion, and PATH:ko04512 ECM-receptor interaction predicted by LncADeep are also reasonable, as these pathways are related with cancers. However, IPMiner-predicted proteins again give only two over-represented KEGG pathways (PATH:ko05200 Pathways in cancer and PATH:ko04151 PI3K-Akt signaling pathway), which are too general to specify the functions of TUNAR.

The above case studies showed that LncADeep can give informative functional annotations which comply with the known functions of lncRNAs. For some lncRNAs whose functions remain unclear, LncADeep can still infer their functions and provide helpful hints for biologists.

3.4 Discussion

LncADeep predicts lncRNA-protein interactions, outperforming state-of-the-art tools and showing several advantages. For instance, LncADeep uses mRMR for feature selection and keeps only the most discriminative features, which helps to reduce overfitting. Besides, LncADeep integrates both sequence and structure features, which is more robust than using only one kind of feature. Furthermore, LncADeep is a user-friendly stand-alone tool that can predict interactions in large scale. In contrast, RPISeq [106], RPI-pred [107] and rpiCool [108] cannot be used for large scale interaction prediction, as RPISeq and RPI-pred provide only online tools and rpiCool can predict only 250 lncRNA-protein pairs in one run. Finally, as a lncRNA functional annotation tool, LncADeep offers annotations automatically for lncRNAs, and case studies have shown that LncADeep can provide helpful functional annotations. Unlike the databases, such as lncRNA2Function [168] and FARNA [169], relying on co-expression profiles, LncADeep gives functional annotations based on only the sequence of lncRNAs and proteins, offering a new direction for annotating the functions of lncRNAs. We expect that researchers can obtain experimental hints by integrating the annotations from LncADeep and other databases (like lncRNA2Function and FARNA).

To benchmark the time cost for predicting lncRNA- protein interactions, we paired one alternative transcript of lncRNA HOTAIR (ENST00000439545) with 20,121 proteins and tested tools on the same machine (Table S9) and measured the elapsed time. As RPISeq [106], RPI-pred [107], and rpiCool [108] were not suitable for large-scale interaction prediction, we benchmarked IPMiner [109], lncPro [86] and LncADeep. It took 4.5 minutes for LncADeep to predict 20,121 lncRNA-protein pairs, while 30 minutes for IPMiner, and 247 minutes for lncPro. It is noteworthy that the time cost for LncADeep includes annotating the functions for lncRNAs with over-represented KEGG and Reactome pathways. Thus, for predicting lncRNA-protein interaction, LncADeep was the fastest among the

tools that can predict interactions in large scale.

However, LncADeep can also suffer some drawbacks on predicting lncRNA-protein interactions like other methods. First, the training dataset is relatively small, which covers only a small amount of interacting lncRNA-protein pairs. To make full use of the capability of deep learning, more training data is required to reveal the hidden interacting mechanisms between lncRNA and protein. Second, generating non-interacting lncRNA-protein pairs by pairing lncRNAs and proteins and randomly choosing can be of risk, as some lncRNA-protein pairs can be potential unverified interacting pairs, which might bias the trained model. However, it is difficult to verify non-interacting lncRNA-proteins, because to some extent, the interaction between lncRNA and protein can only be verified rather than excluded. Therefore, training a model with only interacting pairs can be an alternative strategy to address this problem, such as PRIPU [170], which trained a biased-SVM by adapting its cost function. Finally, the function annotation of lncRNAs might also be biased since it is dependent on the predicted interacting proteins of lncRNAs. Nevertheless, inferring the functions of lncRNAs through its interacting proteins provides an alternative approach to investigate lncRNAs and can help to offer biological insights. In future work, we plan to collect more experimentally verified interacting lncRNA-protein pairs and tackle the problem related to non-interacting pairs. Since LncADeep is readily adapted, we expect to give more informative functional annotations for lncRNAs with additional training datasets.

3.5 Conclusions

With deep learning, we presented a novel method for predicting lncRNA-protein interactions and annotating lncRNA functions, which has also been implemented in our tool LncADeep. In particular, to predict lncRNA-protein interactions, LncADeep first integrates sequence and structure features of lncRNAs and proteins into deep neural networks. Based on the comprehensive performance comparison between LncADeep and other tools, re-

sults showed that LncADeep outperformed state-of-the-art tools, including lncPro, RPISeq, RPI-pred, rpiCool, and IPMiner, for predicting lncRNA-protein interactions. Since accurate lncRNA-protein interactions can help to infer the functions of lncRNAs, we then employ LncADeep to annotate the functions for the 27,384 lncRNAs collected in GENCODE Release 24. With the interacting proteins of lncRNAs, LncADeep infers the functions of these lncRNAs through KEGG and Reactome pathway enrichment analysis and functional module detection. Consequently, LncADeep inferred 702,675 associations with 140 KEGG pathways and 1,839,272 associations with 422 Reactome pathways for the 27,384 lncRNAs, with an average of 25 KEGG and 67 Reactome pathways associated with each lncRNA, conforming with the complexity of lncRNA functions. Besides, case studies on well-studied lncRNAs showed that LncADeep's functional annotations for lncRNAs accord with their known functions, indicating LncADeep can give helpful information for the functions of lncRNAs. Last but not least, the performance of LncADeep demonstrated the effectiveness of deep learning methods, which are capable of learning sophisticated hidden structures in data. We expect that LncADeep can offer informative functional annotations for lncRNAs, and then facilitate the understanding of associations among lncRNAs, gene regulations and diseases.

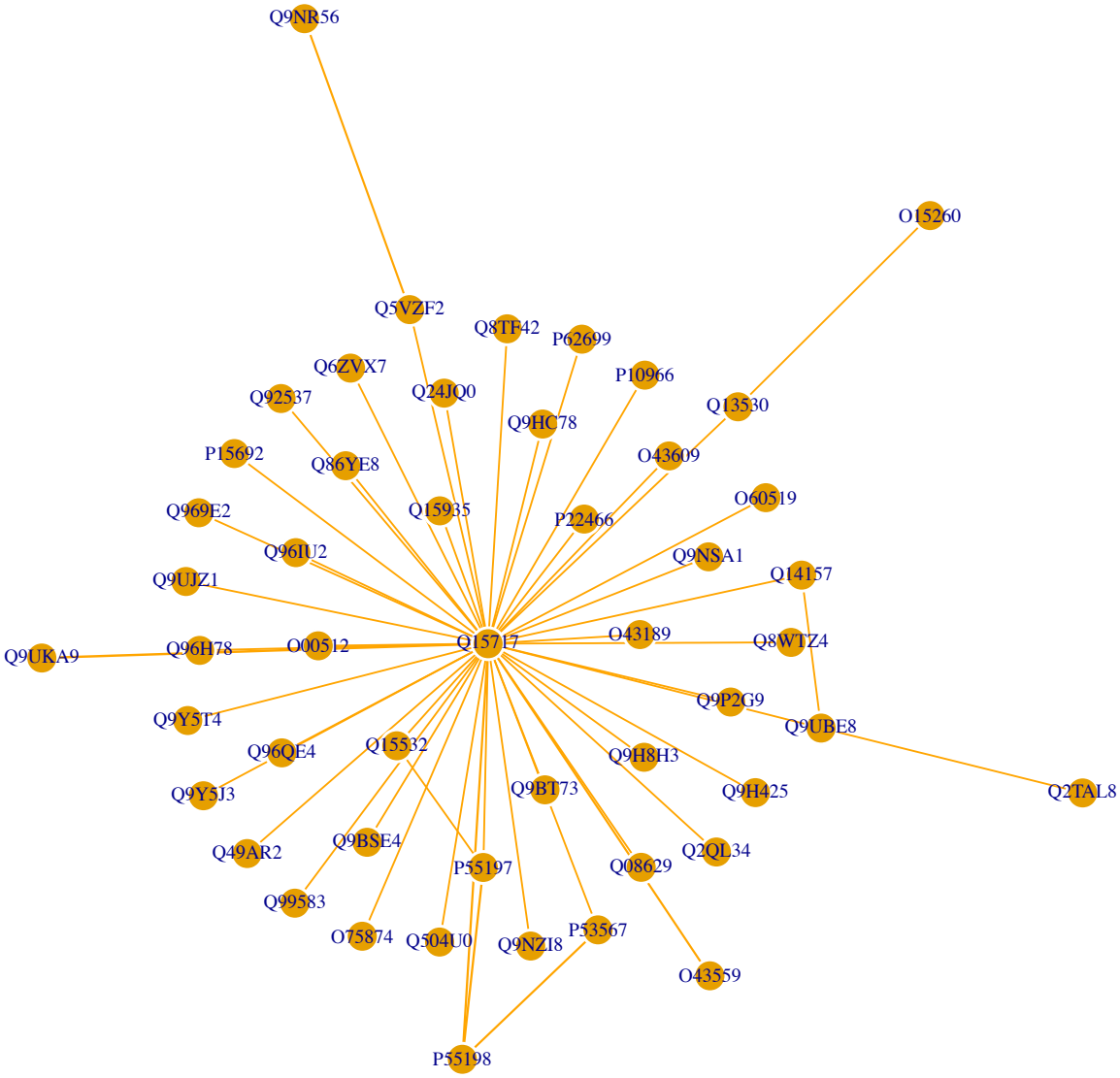


Figure 3.5: An example of the detected functional module.

This is an example functional module detected for one alternative transcript of the lncRNA HOTAIR. Functional module are expected to help to provide useful information for biologist with further interpretation.

CHAPTER 4
APPLICATION OF LNCADEEP AND CONCLUDING REMARKS OF THIS DISSERTATION

As an illustration for the application of LncADeep, this chapter describes the steps to identify and functionally annotate lncRNAs from an RNA-seq dataset using LncADeep. Besides, the conclusions from Appendices A and B are used for selecting a better RNA-seq aligner. In the end, we give the concluding remarks of this dissertation.

4.1 Introduction

As a common category of non-invasive precursor lesion to breast, ductal carcinoma in situ (DCIS) can evolve to invasive breast cancer, especially high-grade DCIS (HG-DCIS) [171, 172]. To investigate the molecular portrait of HG-DCIS, *Abba, Martin C., et al.* conducted transcriptome analysis of pure high-grade DCIS (HG-DCIS) and normal breast epithelial samples [172], and observed that lncRNA HOTAIR was upregulated in HG-DCIS. However, the transcriptome analysis in [172] focused on annotated transcripts and did not reveal novel transcripts associated with HG-DCIS. To find if there are any novel transcripts potentially related to HG-DCIS, especially novel lncRNAs, we downloaded their RNA-seq dataset and conducted the lncRNA identification and annotation experiment. In particular, this experiment is intended to illustrate how to incorporate the tool (i.e., LncADeep) proposed in this dissertation into RNA-seq data analysis.

4.2 Materials and methods

The workflow of the experiment is illustrated in Figure 4.1. To choose the better alignment results for the subsequent analysis, we use two aligners, HISAT2 [17] and STAR [15]. We

use StringTie [32] for sequence assembly and expression quantification, and Ballgown [43] for differential expression analysis, as suggested in [26]. In the end, we use LncADeep for lncRNA identification and annotation.

4.2.1 Dataset

Abba, Martin C. et. al. provided a molecular portrait of high-grade ductal carcinoma by exome capture sequencing analysis, reduced representation bisulfite sequencing analysis and RNA sequencing analysis [172]. Since we focus on RNA-seq data analysis, we use the RNA-seq dataset as provided in [172] with accession number GSE69994. In particular, we choose ten samples each for normal breast epithelium and HG-DCIS samples (Table 4.1), which were sequenced by Illumina HiSeq2000 platform (76-nt paired-end reads and about 40 million tags per sample). The detailed description of the samples can be referred to in [172].

4.2.2 RNA-seq data analysis

To select a better aligner for sequence alignment, we used two aligners, HISAT2 and STAR, which belong to the fastest aligners. For reference genome and genome annotation, we utilized UCSC hg 38 and GENCODE Release 26. After aligning the short reads to the reference genome, we computed two metrics, ReadsAlignedPercentage and ZeroMismatchPercentage, to profile the sequence alignment performance. According to the conclusions of Appendices A and B, the aligner that produce higher ReadsAlignedPercentage and ZeroMismatchPercentage will be a better choice for gene expression estimation and differential expression analysis. Therefore, the alignment results generated by the better aligner were used for the subsequent analysis.

Then, the alignment results were used for reference-guided sequence assembly and transcript expression quantification using StringTie, where we used UCSC hg38 as reference genome and GENCODE Release 26 as genome annotation. To identify differentially

Table 4.1: Data description

Sample ID	Run accession	Description
1	SRR2040339	Normal
2	SRR2040340	Normal
3	SRR2040341	Normal
4	SRR2040342	Normal
5	SRR2040343	Normal
6	SRR2040344	Normal
7	SRR2040345	Normal
8	SRR2040346	Normal
9	SRR2040347	Normal
10	SRR2040348	Normal
11	SRR2040349	pure HG-DCIS
12	SRR2040350	pure HG-DCIS
13	SRR2040351	pure HG-DCIS
14	SRR2040352	pure HG-DCIS
15	SRR2040353	pure HG-DCIS
16	SRR2040354	pure HG-DCIS
17	SRR2040355	pure HG-DCIS
18	SRR2040356	pure HG-DCIS
19	SRR2040357	pure HG-DCIS
20	SRR2040358	pure HG-DCIS

expressed transcripts between normal breast epithelium and HG-DCIS samples, we used Ballgown.

4.2.3 lncRNA identification and functional annotation

To identify and functionally annotate lncRNAs, we used the tool proposed in this dissertation, LncADeep. In this study, we would like to find novel lncRNAs which are associated with HG-DCIS. Therefore, first of all, we obtained differential expressed transcripts from the results of Ballgown. Then, we filtered the differential expressed transcripts which have been annotated in GENCODE Release 26, and kept unannotated transcripts for lncRNA identification using LncADeep. Finally, we used LncADeep to predict interacting proteins and infer functions for the novel lncRNAs.

4.3 Results

For sequence alignment, HISAT2 consistently outperformed STAR with higher ReadsAlignedPercentage (Figure 4.2) and ZeroMismatchPercentage (Figure 4.3). For HISAT2, the ReadsAlignedPercentage of all samples were above 97%, while for STAR, the ReadsAlignedPercentage of all samples ranged from 95% to 98%, which decreased the ReadsAlignedPercentage by an additional $\sim$3%, indicating that most of the reads can be aligned. However, when comparing ZeroMismatchPercentage, HISAT2 showed higher advantage than STAR, increasing the ZeroMismatchPercentage by an additional $\sim$8%. According to the conclusions drawn from Appendices 2 and 3, we chose the alignment results generated by HISAT2 for the subsequent analysis.

Using Ballgown (q-value less than 0.01), we found 2,680 differentially expressed transcripts between normal and HG-DCIS samples, 66.8% (1,789) were annotated transcripts and 33.2% (891) were novel transcripts. Among the annotated transcripts, 93.4% were protein-coding transcripts and 6.6% were noncoding RNAs. To identify lncRNAs from the novel transcripts, we used LncADeep and identified 412 novel lncRNAs. In HG-DCIS samples, we found that 88% of annotated lncRNAs were over expressed (Figure 4.4), while about 79% novel lncRNAs were over expressed (Figure 4.5). Among the annotated lncRNAs, we noticed that some had been identified as potentially associated with breast cancer, such as RP11-111M22.4 [173], RP3-393E18.2 [173], and UCA1 [174]. Besides, we plotted the expression level of six novel lncRNAs (Figure 4.6), which are the most significant differentially expressed lncRNAs. The gtf annotation for these novel lncRNAs is list as Figure 4.7. After identifying the novel lncRNAs, we then used LncADeep to predict their interacting proteins and then infer their functions. As an example, we list the inferred KEGG pathways for the novel lncRNA MSTRG.33791.3, where LncADeep successfully predicted that MSTRG.33791.2 could be relevant to PATH:ko05224 Breast cancer (Figure 4.8).

In this experiment, we incorporated the proposed tool (i.e., LncADeep) in this dissertation and conclusions drawn from Appendices A and B for RNA-seq data analysis. First, according to the conclusions of Appendices A and B, for sequence alignment, the alignment results with higher ReadsAlignedPercentage and ZeroMismatchPercentage were used for the subsequent analysis, which tend to yield more accurate analysis results. Second, as a state-of-the-art tool for lncRNA identification and functional annotation, LncADeep was used to identify novel lncRNAs and annotate their functions. In particular, we found 412 novel lncRNAs which can be potentially associated with HG-DCIS, and the function annotations for these lncRNAs also correlated them to breast cancer, to which HG-DCIS can evolve. Although further verification for these lncRNAs and their function annotations are needed, this experiment demonstrated that LncADeep could help to identify and functionally annotate novel lncRNAs and provide helpful hints for biologists.

4.4 Concluding remarks of this dissertation

This dissertation proposed LncADeep (Figure 4.9), an *ab initio* lncRNA identification and annotation tool based on deep learning methods.

LncADeep is composed of two parts, the first part is to identify lncRNA (Figure 4.9A) based on a deep learning algorithm, deep belief networks. For lncRNA identification, we integrated both intrinsic and homology features for model construction. In particular, we constructed two models, one targeting full-length transcripts, and the other targeting transcripts including full- and partial-length. The model targeting full-length transcripts will be helpful when the third generation sequencing technologies, which suffice for long read sequencing, becomes prevalent. Meanwhile, in current RNA-seq dataset, the assembled transcripts from the short reads are composed of full- and partial-length, while most lncRNA identification tools focus on identifying lncRNA from full-length transcripts. Therefore, we designed a model targeting transcripts including full- and partial-length. Specifically, we proposed LCDS-based model to address the problem about locating CDS from partial-

length mRNAs, and majority voting to improve the generalization ability. Results showed that LncADeep outperformed state-of-the-art lncRNA identification tools, including CPC, CPC2, CPAT, CNCI, COME, lncRScan-SVM, longdist, lncRNA-MFDL, lncScore, FEElnc and PLEK, on full-length transcripts (accuracy of 97.7%), as well as transcripts including full- and partial-length (accuracy of 94.2%). Besides, as a reference-free (or *ab initio*) tool, LncADeep still outperform other tools on cross-species lncRNA identification on full-length transcripts (accuracy of 96.7%) and transcripts of full- and partial-length (94.2%).

The second part is to predict lncRNA-protein interactions and infer lncRNA functions (Figure 4.9B). To functionally annotate lncRNAs, we developed a method for predicting lncRNA-protein interactions, and then used the predicted interacting proteins for inferring the functions of lncRNAs. Through integrating both sequence and structure information into a deep neural network for predicting lncRNA-protein interaction, LncADeep outperformed stat-of-the-art tools (including RPISeq, RPI-pred, rpiCool, lncPro, and IPMiner) with an accuracy of 90.8%. With the predicted interacting proteins of lncRNAs, we then employed LncADeep to annotate the functions for the 27,384 lncRNAs collected in GENCODE Release 24, through KEGG and Reactome pathway enrichment analysis, and functional module detection. Consequently, LncADeep inferred an average of 25 KEGG and 67 Reactome pathways associated with each lncRNA, according with the complexity of lncRNA functions. In addition, examples on well-studied lncRNAs showed that LncADeep's functional annotations for lncRNAs comply with their known functions.

To illustrate the steps to identify and functionally annotate lncRNAs from a RNA-seq dataset using LncADeep, we conducted a lncRNA identification and functional annotation experiment using a public RNA-seq dataset of high-grade ductal carcinoma in situ (HG-DCIS), which can evolve to invasive breast cancer. In particular, using LncADeep, we identified 412 differentially expressed novel lncRNAs which can be involved with HG-DCIS, and the function annotations for these lncRNAs also associated them with pathways related to breast cancer, to which HG-DCIS can evolve.

In summary, we proposed LncADeep (http://cqb.pku.edu.cn/ZhuLab/LncADeep), which is, to our best knowledge, the first tool that can identify lncRNAs and automatically provide functional annotations for lncRNAs. We expect that not only can LncADeep contribute to identifying lncRNAs, but also provide helpful functional information for investigating the associations among lncRNAs, gene regulation and diseases, and then facilitate the large-scale automatic genome annotation.

In future work, we plan to collect more experimentally verified interacting lncRNA-protein pairs and further improve the performance of predicting lncRNA-protein interaction, which will in turn improve the inference of lncRNA functions. Since LncADeep is readily adapted, we expect to give more informative functional annotations for lncRNAs with additional training datasets. Besides, we plan to implement a webserver integrating lncRNA identification and functional annotation, and displaying the predicted functions of lncRNAs.

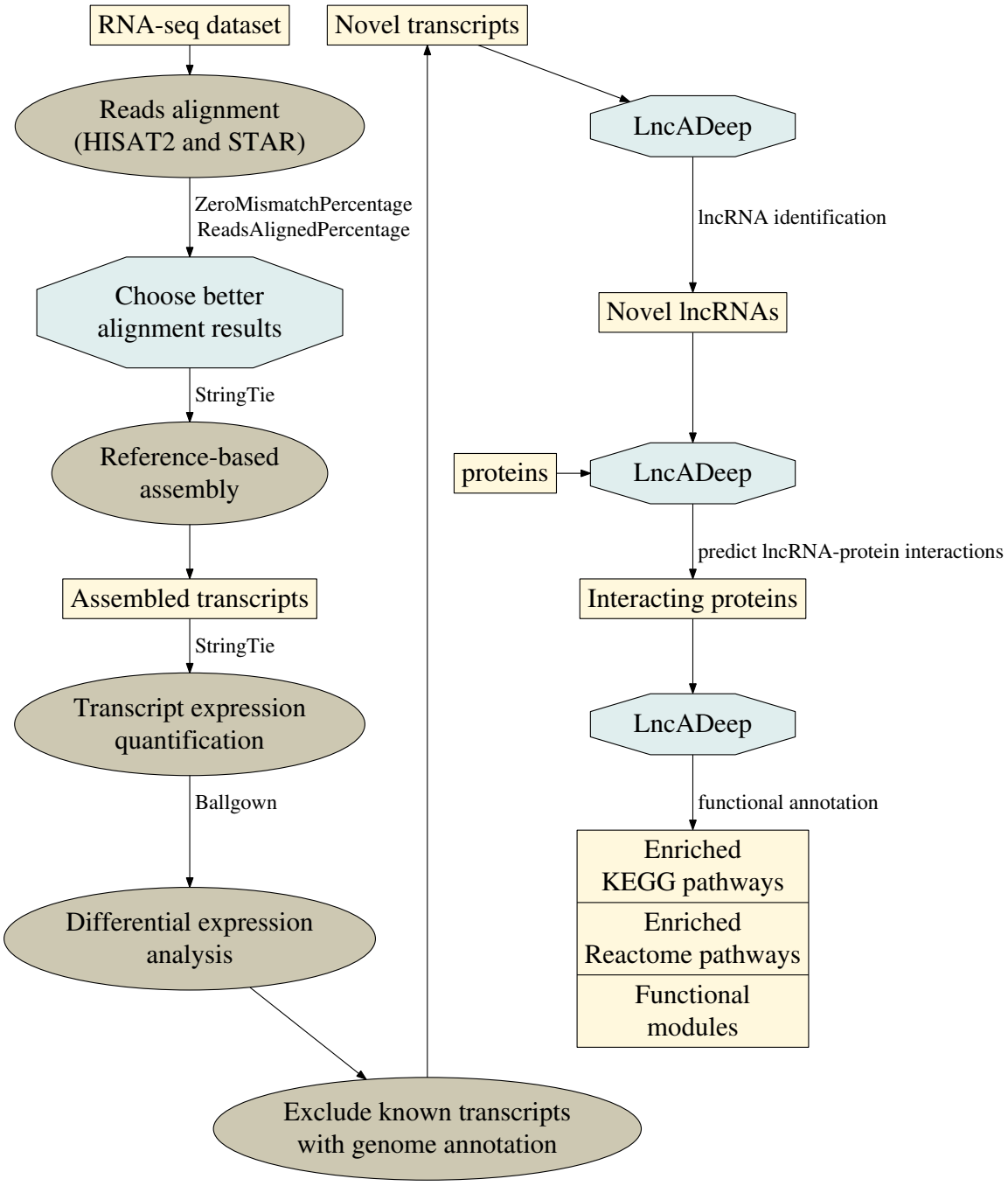


Figure 4.1: The flowchart of the lncRNA identification and annotation experiment.

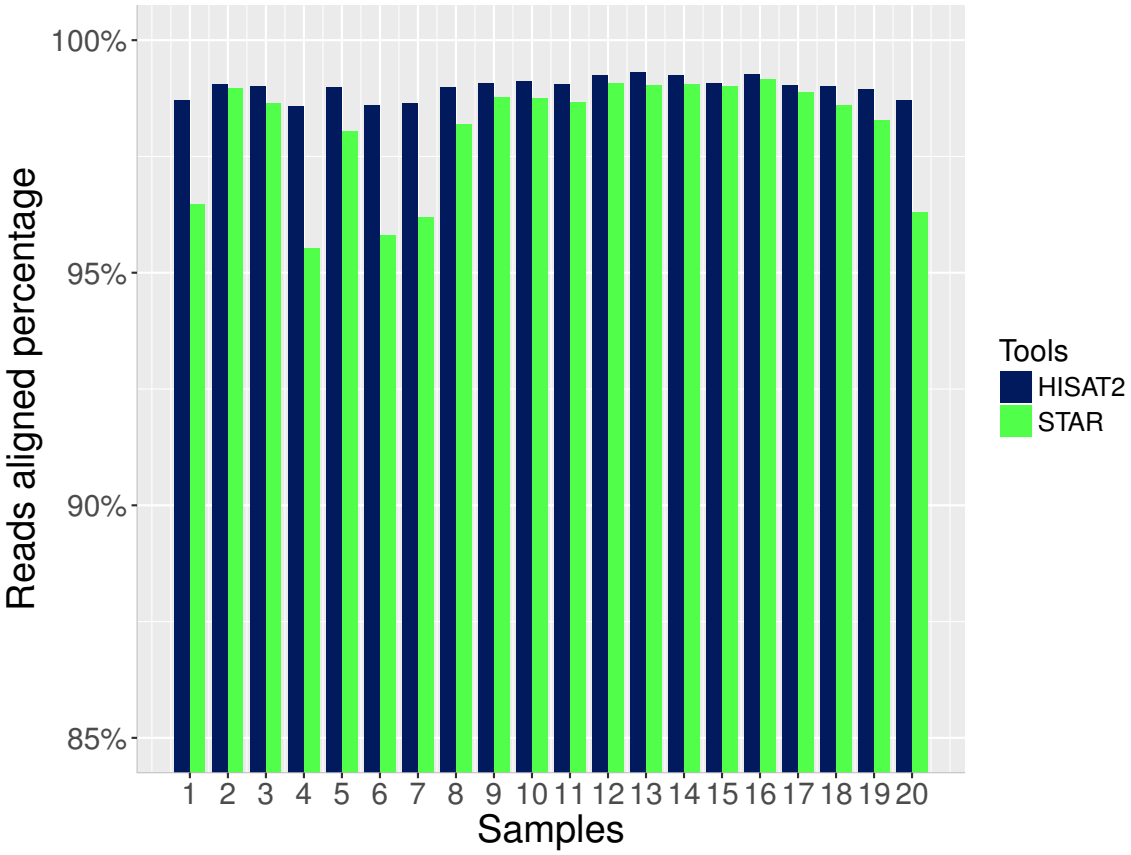


Figure 4.2: The performance of alignment results (ReadsAlignedPercentage).

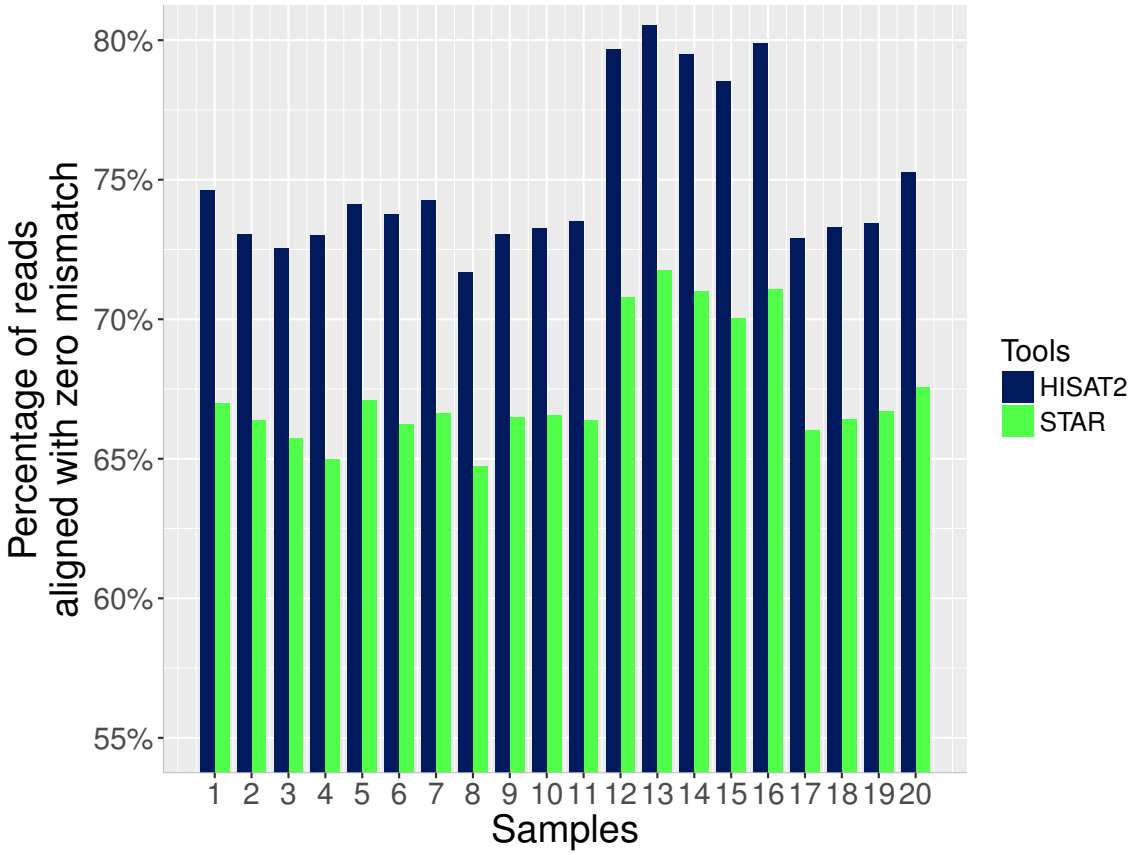


Figure 4.3: The performance of alignment results (ZeroMismatchPercentage).

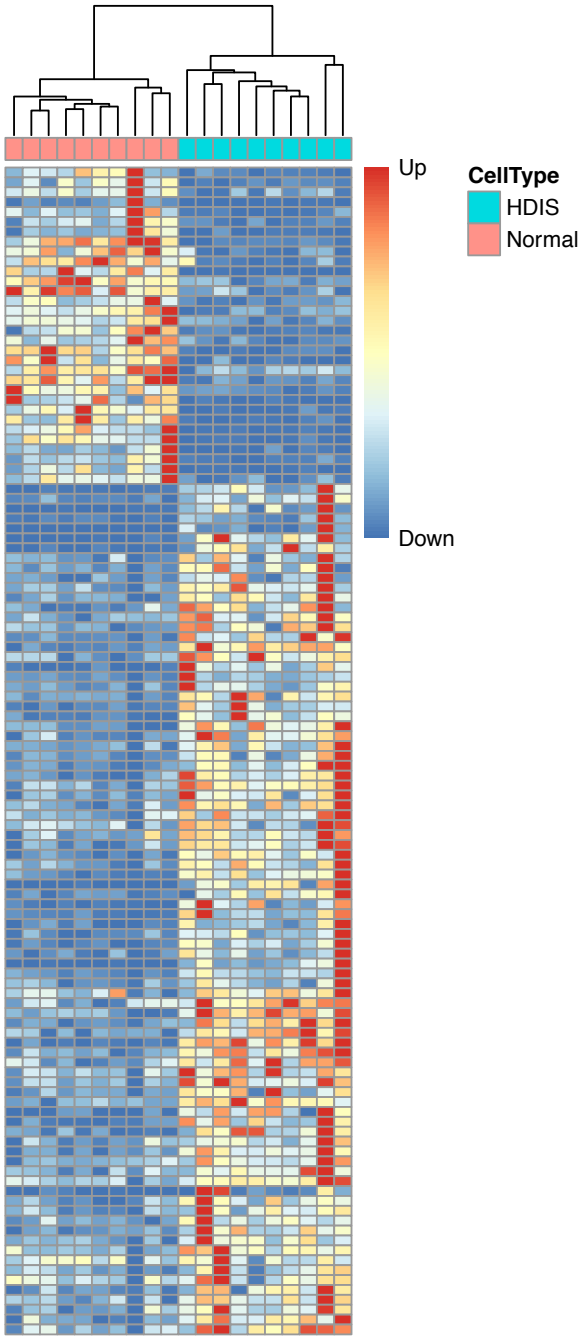


Figure 4.4: Differentially expressed known lncRNAs between normal and HG-DCIS samples.

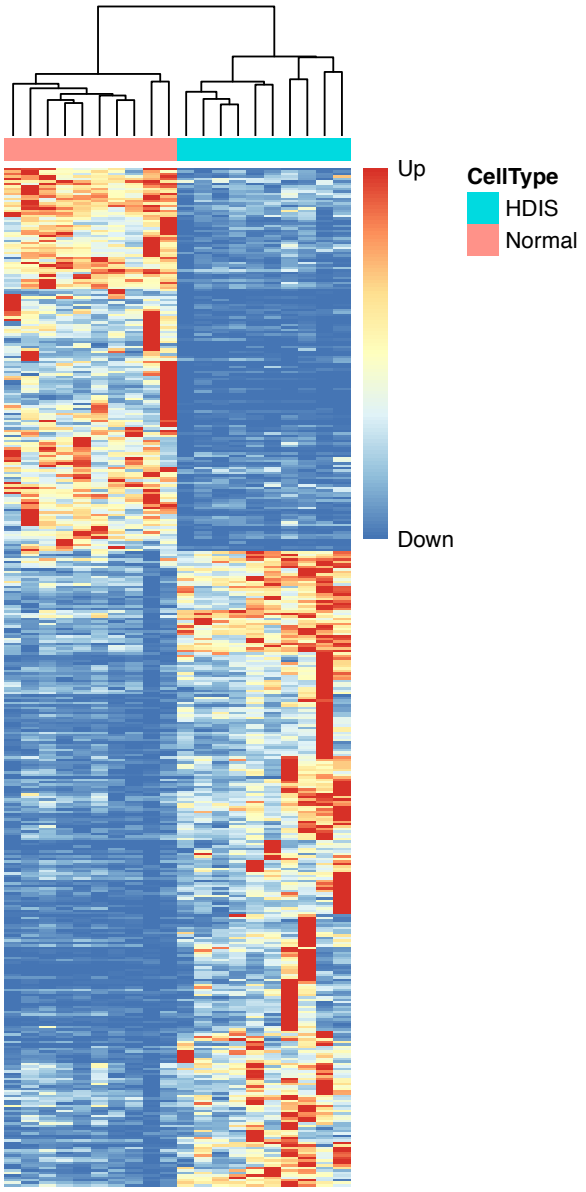


Figure 4.5: Differentially expressed novel lncRNAs between normal and HG-DCIS samples.

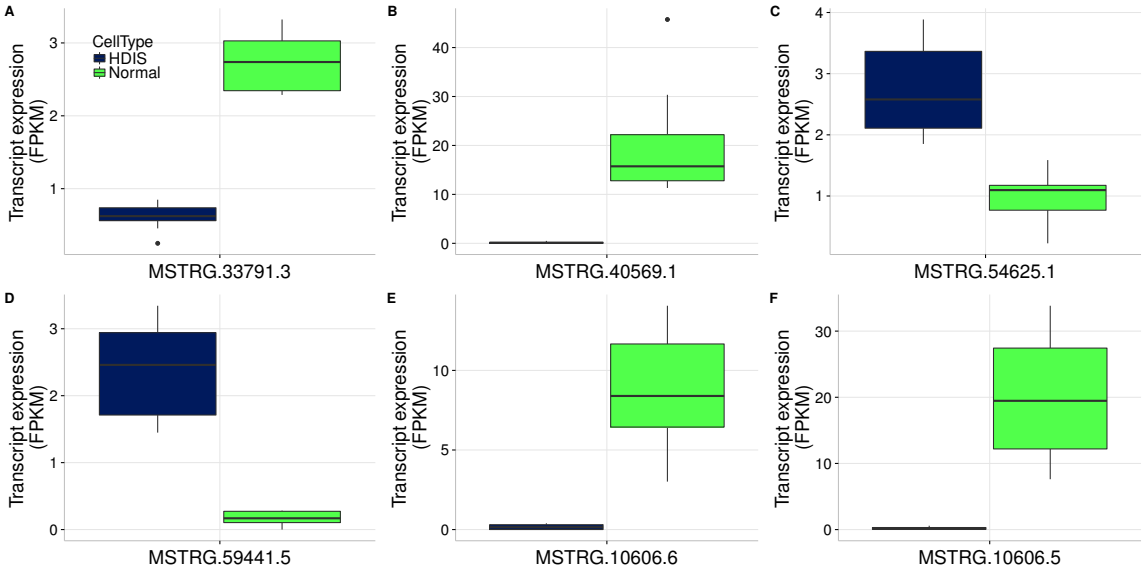


Figure 4.6: The transcript expression of novel differentially expressed lncRNAs.

chr2	StringTie	transcript	47906185	47906529	1000	+	.	gene_id	**MSTRG.33791;**	transcript_id	MSTRG.33791.3;			
chr2	StringTie	exon	47906185	47906529	1000	+	.	gene_id	**MSTRG.33791;**	transcript_id	MSTRG.33791.3;	exon_number	1;	
chr22	StringTie	transcript	30245484	30245841	1000	+	.	gene_id	**MSTRG.40569;**	transcript_id	MSTRG.40569.1;			
chr22	StringTie	exon	30245484	30245841	1000	+	.	gene_id	**MSTRG.40569;**	transcript_id	MSTRG.40569.1;	exon_number	1;	
chr6_KI270798v1_alt	StringTie	transcript	141729	143558	1000	+	.	gene_id	**MSTRG.54625;**	transcript_id	MSTRG.54625.1;			
chr6_KI270798v1_alt	StringTie	exon	141729	143558	1000	+	.	gene_id	**MSTRG.54625;**	transcript_id	MSTRG.54625.1;	exon_number	1;	
chr8	StringTie	transcript	94957999	94989462	1000	+	.	gene_id	**MSTRG.59441;**	transcript_id	MSTRG.59441.5;			
chr8	StringTie	exon	94957999	94958179	1000	+	.	gene_id	**MSTRG.59441;**	transcript_id	MSTRG.59441.5;	exon_number	1;	
chr8	StringTie	exon	94980855	94980973	1000	+	.	gene_id	**MSTRG.59441;**	transcript_id	MSTRG.59441.5;	exon_number	2;	
chr8	StringTie	exon	94988358	94989462	1000	+	.	gene_id	**MSTRG.59441;**	transcript_id	MSTRG.59441.5;	exon_number	3;	
chr11	StringTie	transcript	65892645	65900419	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.6;			
chr11	StringTie	exon	65892645	65893296	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.6;	exon_number	1;	
chr11	StringTie	exon	65894014	65894116	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.6;	exon_number	2;	
chr11	StringTie	exon	65896809	65900419	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.6;	exon_number	3;	
chr11	StringTie	transcript	65892645	65900419	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.5;			
chr11	StringTie	exon	65892645	65893296	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.5;	exon_number	1;	
chr11	StringTie	exon	65894014	65897006	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.5;	exon_number	2;	
chr11	StringTie	exon	65900241	65900419	1000	-	.	gene_id	**MSTRG.10606;**	transcript_id	MSTRG.10606.5;	exon_number	3;	

Figure 4.7: The gtf annotation of novel differentially expressed lncRNAs.

KEGG_path_ID	KEGG_pathway	p_value	adj_p_value
PATH:ko04512	ECM-receptor interaction	1.90E-10	5.20E-09
PATH:ko05224	Breast cancer	3.20E-05	4.30E-04
PATH:ko05217	Basal cell carcinoma	1.00E-04	8.80E-04
PATH:ko04510	Focal adhesion	1.30E-04	8.80E-04
PATH:ko04916	Melanogenesis	3.80E-04	1.90E-03
PATH:ko04310	Wnt signaling pathway	4.90E-04	1.90E-03
PATH:ko04390	Hippo signaling pathway	4.90E-04	1.90E-03
PATH:ko04151	PI3K-Akt signaling pathway	3.00E-03	1.00E-02
PATH:ko04150	mTOR signaling pathway	4.40E-03	1.30E-02
PATH:ko00360	Phenylalanine metabolism	1.10E-02	2.90E-02
PATH:ko00680	Methane metabolism	1.30E-02	3.30E-02
PATH:ko05200	Pathways in cancer	1.60E-02	3.40E-02
PATH:ko05205	Proteoglycans in cancer	1.50E-02	3.40E-02
PATH:ko04016	MAPK signaling pathway - plant	1.80E-02	3.40E-02
PATH:ko05133	Pertussis	2.60E-02	4.40E-02
PATH:ko04330	Notch signaling pathway	2.90E-02	4.40E-02
PATH:ko04974	Protein digestion and absorption	2.90E-02	4.40E-02
PATH:ko04550	Signaling pathways regulating pluripotency of stem cells	2.60E-02	4.40E-02
PATH:ko05144	Malaria	3.20E-02	4.50E-02
PATH:ko04610	Complement and coagulation cascades	4.00E-02	4.50E-02
PATH:ko05150	Staphylococcus aureus infection	4.00E-02	4.50E-02
PATH:ko01210	2-Oxocarboxylic acid metabolism	3.70E-02	4.50E-02
PATH:ko04933	AGE-RAGE signaling pathway in diabetic complications	3.80E-02	4.50E-02
PATH:ko04145	Phagosome	3.40E-02	4.50E-02
PATH:ko04971	Gastric acid secretion	4.60E-02	4.70E-02
PATH:ko04340	Hedgehog signaling pathway	4.70E-02	4.70E-02
PATH:ko00532	Glycosaminoglycan biosynthesis - chondroitin sulfate / dermatan sulfate	4.60E-02	4.70E-02

Figure 4.8: The KEGG pathway annotation of a novel differentially expressed lncRNA MSTRG.33791.3.

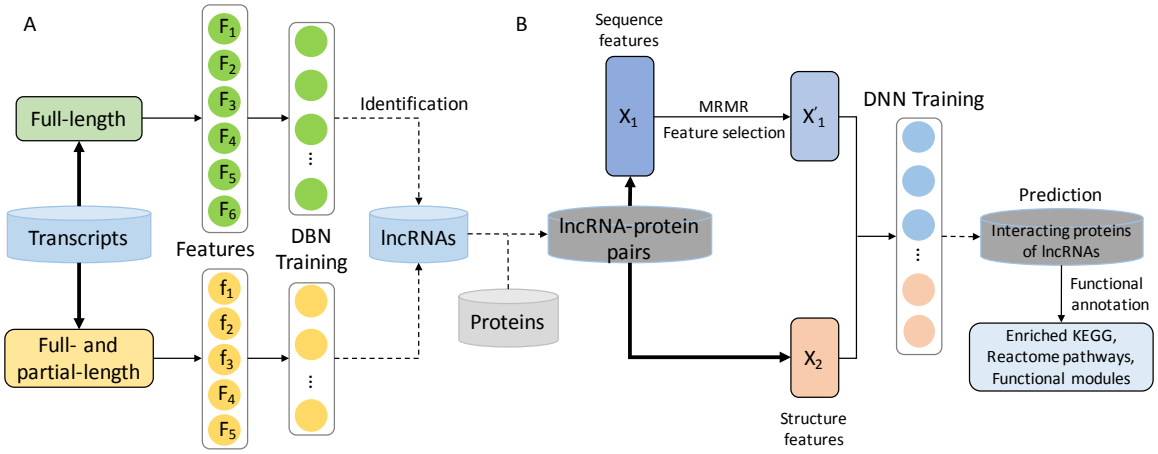


Figure 4.9: The flowchart of LncADeep.

(A) We construct two models for lncRNA identification, one for full-length transcripts, and the other for transcripts including full- and partial-length. F_1 to F_6 refer to ORF length and coverage, the EDP of ORF, Mean hexamer score, Fickett nucleotide features, HMMER index, and UTR length and GC content, respectively. f_1 to f_3 represent LCDS length and coverage, the EDP of LCDS, and Mean hexamer score, respectively. **(B)** We use sequence and structure features for the prediction of lncRNA-protein interaction. X_1, $X_1^{'}$ and X_2 refer to sequence features, sequence features after feature selection, and structure features, respectively. The identified lncRNAs can be input for predicting lncRNA-protein interactions, then the interacting proteins can be used for inferring the functions of lncRNAs.

Appendices

APPENDIX A

THE IMPACT OF RNA-SEQ ALIGNERS ON DEG DETECTION

Components of this chapter have been published as: Yang, C., Wu, P. Y., Phan, J. H., & Wang, M. D. (2014, December). The impact of RNA-seq alignment pipeline on detection of differentially expressed genes. In Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on (pp. 1376-1379). IEEE. DOI: 10.1109/GlobalSIP.2014.7032351

A.1 Abstract

RNA-seq data analysis pipelines are generally composed of sequence alignment, expression quantification, expression normalization, and differentially expressed gene (DEG) detection. Each step has numerous specific tools or algorithms, so we cannot explore all combinatorial pipelines and provide a comprehensive comparison of pipeline performance. To understand the mechanism of RNA-seq data analysis pipelines and provide some useful information for pipeline selection, we believe it is necessary to analyze the interactions among pipeline components. In this paper, by combining different alignment algorithms with the same quantification, normalization, and DEG detection tools, we construct nine RNA-seq pipelines to analyze the impact of RNA-seq alignment on downstream applications of gene expression estimates. Specifically, we find moderate linear correlation between the number of DEGs detected and the percentage of reads aligned with zero mismatch.

A.2 Introduction

Facilitated by next-generation sequencing (NGS) technology, high-throughput RNA sequencing (RNA-seq) interrogates the comprehensive profile of transcriptomes [4], enabling detailed identification of gene isoforms, translocation events, nucleotide variations, and post-transcriptional base modifications [175, 3].

A standard RNA-seq data analysis pipeline consists of (1) sequence read mapping, (2) expression quantification, (3) expression normalization, and (4) differentially expressed gene (DEG) detection, and each step has a considerable number of bioinformatics tools. Since a pipeline consists of a sequence of the selected tools from each step, the combination of these tools provides a number of choices, yet raises the following question: Which pipeline should we use? Intuitively, the best pipeline would be composed of the best tool in each step. Researchers have conducted comparative analyses for the sequence alignment [176], expression quantification [177], expression normalization, and DEG detection [175] tools. The evaluation of the tools in a pipeline may be informative for pipeline selection. Based on this evaluation, we might select the most accurate alignment, quantification, normalization, and DEG detection tools to construct a pipeline. However, the combination of the best tools does not ensure an accurate analysis result, especially when the performance of the tool is sample-related. For instance, Grant et al. [176] found that the base-level accuracy of alignment pipelines varies among samples. Until now, few studies systematically compared the performance of RNA-seq pipelines. Therefore, it remains uncertain whether the combination of best tools will produce a better-performing pipeline. To provide helpful information for pipeline selection and understand the mechanism of RNA-seq data analysis pipelines, we believe it is necessary to investigate the associations among the steps in RNA-seq pipelines. Once we know how the alignment step affects the final results (e.g., DEG detection), we can determine which alignment tool we should use and even estimate the number of DEGs with alignment metrics that can profile the alignment results.

In this paper, we analyze the impact of RNA-seq alignment pipeline on downstream applications of gene expression estimates, e.g., DEG detection. The rest of this paper is organized as follows. Section II introduces the experimental design and data analysis. Section III discusses the results and the potential impact of alignment on gene expression estimates. Finally, Section IV concludes our work.

A.3 Methodology

The workflow of this study is shown in Figure A.1. To analyze the impact of alignment on gene expression estimates, we vary the alignment tools (Bowtie2 [20], BWA [178], GSNAP [179], Novoalign [180], and WHAM [22]) while using a fixed quantification tool (RSEM [31]), a normalization algorithm (trimmed mean of M-values normalization, TMM [45]), and a DEG detection tool (edgeR [42]).

A.3.1 Dataset

The dataset consists of SEQC samples A and B [181], which contain Stratagenes Universal Human Reference RNA and Ambions Human Brain Reference RNA, respectively. The samples were sequenced with the Illumina HiSeq 2000 platform at three official sequencing sites, including the Beijing Genomics Institute (BGI), the Weill Cornell Medical College (CNL) and the Mayo Clinic (MAY). In this paper, we use only the data sequenced at BGI, which includes four replicates with around five million paired-end reads for each replicate. Each replicate has sixteen lanes, and we use the first two lanes.

A.3.2 Sequence Mapping and Expression Quantification

To analyze the impact of alignment on gene expression estimates, we vary the alignment tools, including Bowtie2, BWA, GSNAP, Novoalign and WHAM. For Bowtie2, GSNAP, Novoalign, and WHAM, we use two sequence alignment reporting strategies, single-hit and multiple-hit. Whereas single-hit aligners report only one location for a single read,

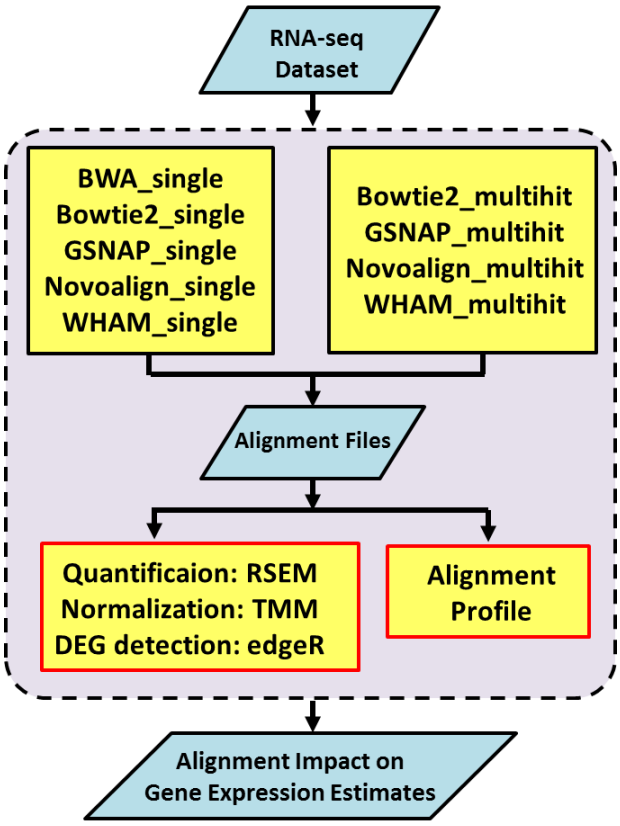


Figure A.1: The workflow for investigating the association between RNA-seq alignment profiles and gene expression estimates.

multiple-hit aligners can report more than one location. BWA only reports single-hit alignments. We use the same reference genome (i.e., UCSC hg19) and the same genome annotation (i.e., AceView [182]) for all alignment pipelines. For gene expression quantification, we use RSEM with both the AceView transcriptome [182] and hg19 as reference genomes. The data generated from RSEM are in the form of gene counts.

A.3.3 Alignment Profiles

We characterize alignment profiles by using the percentage of reads aligned with zero and one mismatch as alignment metrics. Reads aligned with zero or one mismatch are more likely to account for gene expression estimates. We extract the percentage of reads aligned with no mismatch denoted as ZeroMismatchPercentage, and those with at most one mis-

match denoted by OneMismatchPercentage. In addition, we count the number of reads aligned with single- or multiple-hit reporting. Since each sample has four replicates, we first compute the alignment metrics for each replicate, and then calculate the average as the alignment metrics of the sample.

A.3.4 DEG Detection Specificity

For gene expression estimates, evaluating every gene is not possible, especially when most genes have similar expression. As a result, we propose to use DEG detection as a downstream evaluation of gene expression estimates. We identify DEGs using the edgeR package in R. Before detecting DEGs, we use TMM (trimmed mean of M-values normalization) to normalize the data. Since each sample has four replicates (Replicates 1, 2, 3, and 4), we compare two replicates with the other two to detect DEGs (i.e., Replicates 1 and 2 vs. Replicates 3 and 4, Replicates 1 and 3 vs. Replicates 2 and 4, and Replicates 1 and 4 vs. Replicates 2 and 3). With various combinations, we have three groups, that is, we can get three DEG numbers for each sample. Because replicates come from the same sample, ideally the number of DEGs should be close to zero based on the assumption that the pipeline performs well. To capture and model this assumption, we define "DEG index" as "each pipeline's total DEG number" to represent the pipeline's quality. That is, for each pipeline, we add the three DEG numbers as its DEG index. The DEG index can quantify differences among pipelines. Meanwhile, the only variable in the comparison of pipelines is the alignment tool, which will be the only source of the discrepancy among the DEG indices of the pipelines. To investigate the effects of different DEG adjusted p-value thresholds on our observation, we detected DEGs with different thresholds (from p = 0.01 to 0.1). As larger adjusted p-value thresholds indicate looser constraints for DEGs, we expected more DEGs when we gradually increased the thresholds.

A.4 Results and discussion

Figures A.2 and A.3 show that, for most alignment tools for both Samples A and B, more than 60% of reads aligned with zero mismatch, and over 80% of reads aligned with zero or one mismatch, suggesting that the percentage of zero and one mismatch can cover the majority of reads in the alignment files. For both Samples A and B, alignment pipelines showed almost the same trend in ZeroMismatchPercentage and OneMismatchPercentage, suggesting that ZeroMismatchPercentage and OneMismatchPercentage in the alignment tools might be independent of the samples. We also verified that single-hit alignment pipelines only report one hit for each read; in contrast, multi-hit alignment pipelines can report several hits for some reads (Figures A.4 and A.5).

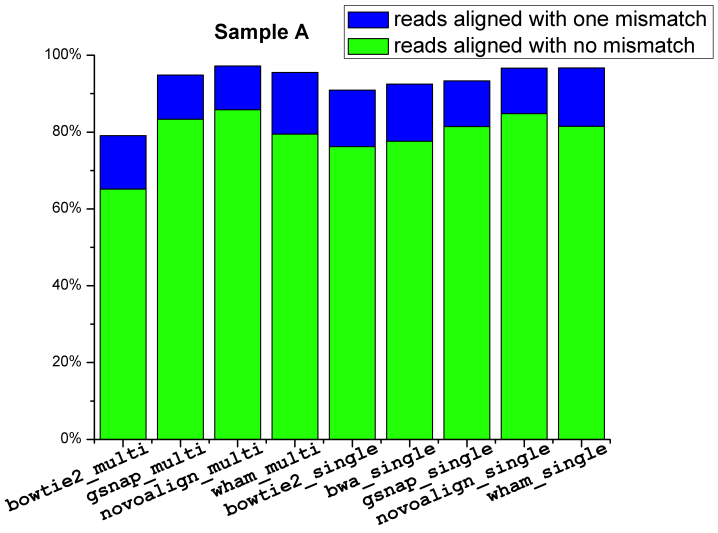


Figure A.2: The alignment profiles of Sample A (percentage of reads aligned with zero or one mismatch).

Figures A.6 and A.7 show the key finding of our study: The DEG indices of RNA-seq pipelines have moderate linear correlation with the percentage of reads aligned with zero or one mismatch (ZeroMismatchPercentage and OneMismatchPercentage). Figures A.6 and A.7 show the impact of alignment pipelines on the DEG indices of Samples A and B, respectively. Note that single- and multiple-hit alignment strategies are distinctive. We

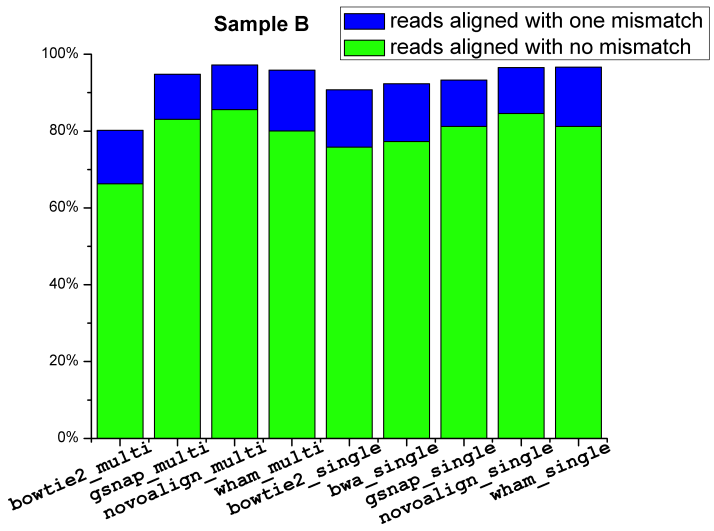


Figure A.3: The alignment profiles of Sample B (percentage of reads aligned with zero or one mismatch).

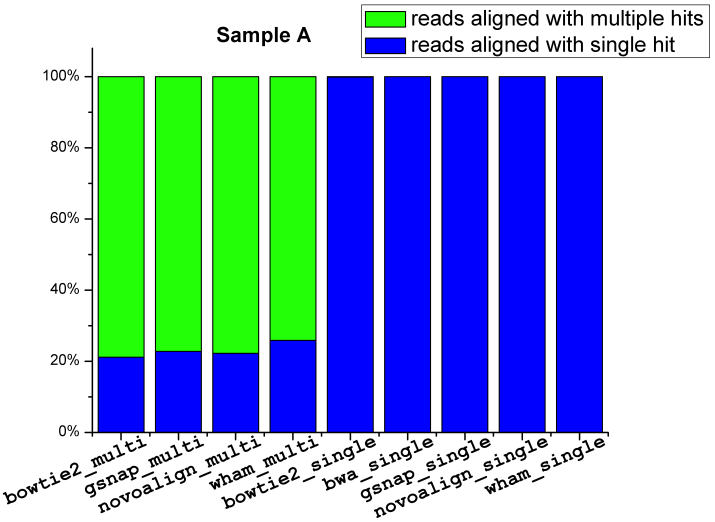


Figure A.4: The alignment profiles Sample A (percentage of reads aligned with single-hit or multiple-hit).

use linear regression to measure their impact on DEG indices separately. For Sample A, both multiple-hit (blue boxes in Figure A.6) and single-hit (red boxes in Figure A.6) DEG indices of alignment pipelines tended to decrease as ZeroMismatchPercentage increased. However, for the OneMismatchPercentage, the correlations between the DEG indices and the alignment pipelines were insignificant (Table A.1). As for Sample B (Figure A.7 and Table A.2), both multiand single-hit DEG indices of the alignment pipeline also had lin-

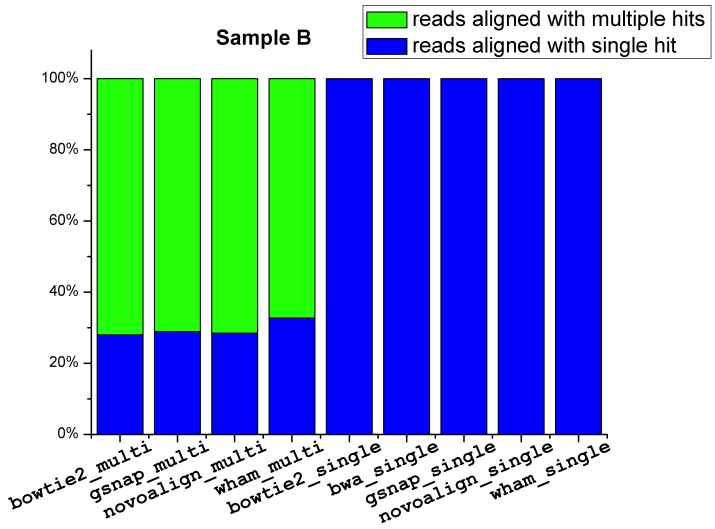


Figure A.5: The alignment profiles of sample B (percentage of reads aligned with single-hit or multiple-hit).

ear correlation with ZeroMismatchPercentage. Unlike that of Sample A, both multi- and single-hit DEG indices of the alignment pipeline exhibited a moderate linear correlation with OneMismatchPercentage in Sample B. This discrepancy might relate to the sample differences. Some sample-related metrics can also account for the impact of alignment pipelines on DEG index apart from the two metrics above. For Sample A, the sample-related metrics might fluctuate among results of alignment pipelines, while for Sample B, the other metrics may be consistent, which leads to that discrepancy. In addition, compared with single-hit alignment algorithms, ZeroMismatchPercentage of multiple-hit alignment algorithms have stronger linear impact on DEG index (Table A.1 and A.2). Overall, our study discovered an alignment pipeline metric ZeroMismatchPercentage with moderate linear impact on gene expression estimation.

A.5 Conclusion

We investigated the impact of alignment pipelines on gene expression estimates of RNA-seq pipelines. First, we constructed nine different RNA-seq pipelines by combining different alignment pipelines with the same quantification, normalization, and DEG detection

Table A.1: Correlation coefficient of Sample A

DEG adjusted p-value threshold	OneMismatchPercentage		ZeroMismatchPercentage	
	multiple-hit	single-hit	multiple-hit	single-hit
0.01	-0.7901	-0.0478	-0.8774	-0.0387
0.02	-0.7518	-0.2361	-0.8842	-0.2792
0.03	-0.8721	-0.2456	-0.9610**	-0.2864
0.04	-0.8537	-0.2841	-0.9363*	-0.3927
0.05	-0.7537	-0.3958	-0.8616	-0.4101
0.06	-0.6826	-0.5797	-0.8091	-0.7022
0.07	-0.7611	-0.5829	-0.8792	-0.7764
0.08	-0.75	-0.3759	-0.8775	-0.6194
0.09	-0.7622	-0.3549	-0.8867	-0.5777
0.1	-0.7023	-0.3952	-0.844	-0.6163

Significance codes: '**' p-value < 0.05, '*' p-value < 0.1.

Table A.2: Correlation coefficient of Sample B

DEG adjusted p-value threshold	OneMismatchPercentage		ZeroMismatchPercentage	
	multiple-hit	single-hit	multiple-hit	single-hit
0.01	-0.5828	-0.3632	-0.7507	-0.7059
0.02	-0.7194	-0.553	-0.8585	-0.8246*
0.03	-0.7689	-0.698	-0.894	-0.8754*
0.04	-0.8104	-0.6447	-0.9208*	-0.8109*
0.05	-0.8026	-0.7132	-0.9142*	-0.8697*
0.06	-0.8234	-0.7764	-0.9175*	-0.9187**
0.07	-0.8689	-0.7473	-0.9452*	-0.9363**
0.08	-0.8546	-0.7435	-0.9315*	-0.9475**
0.09	-0.8499	-0.6872	-0.9345*	-0.9246**
0.1	-0.8091	-0.6547	-0.9118*	-0.9034**

Significance codes: '**' p-value < 0.05, '*' p-value < 0.1.

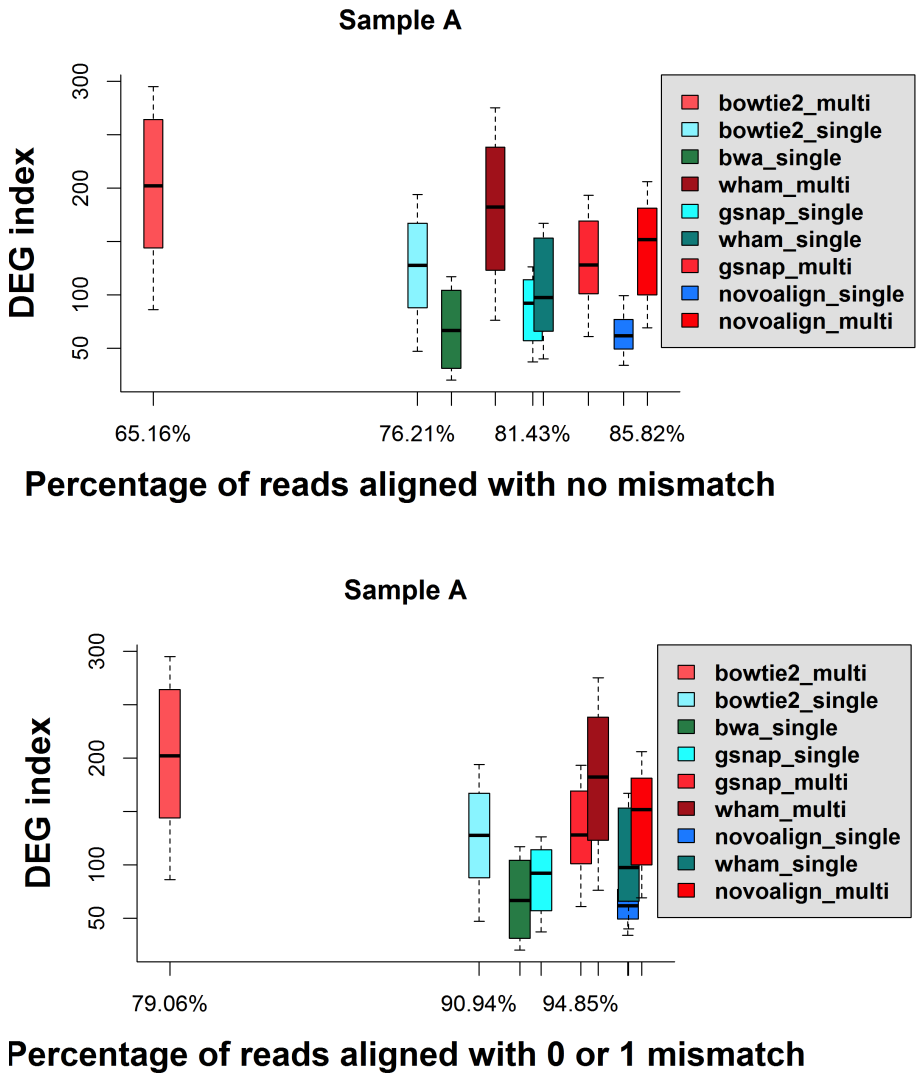
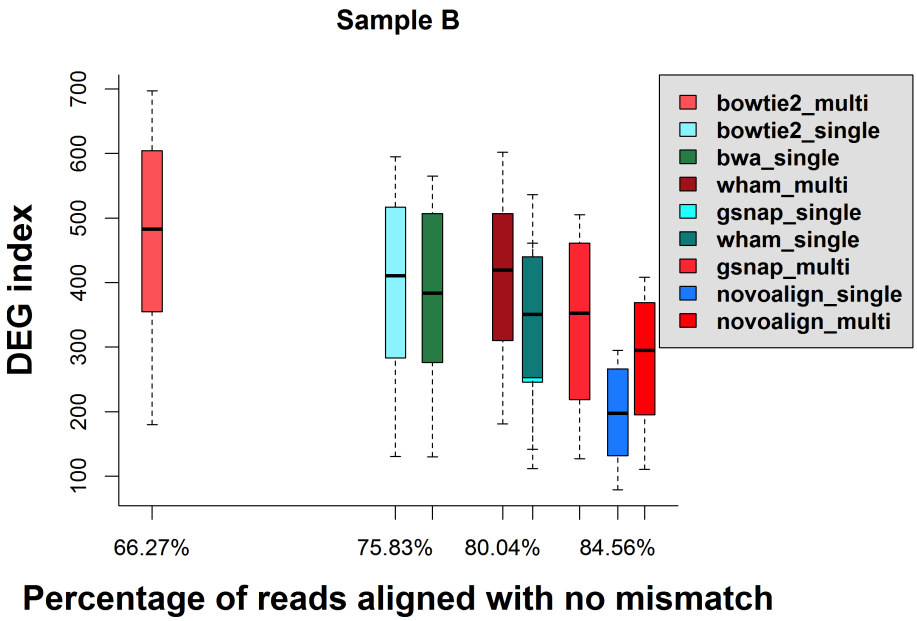


Figure A.6: The impact of alignment pipelines on gene expression estimation (Sample A).

tools. With these RNA-seq pipelines, we computed DEG indices for real datasets. Then, to profile alignment pipelines, we calculated the percentages of reads aligned with zero and one mismatch. Our study indicated that the ZeroMismatchPercentage of alignment pipelines had moderate linear impact on DEG index. Thus, we recommend constructing RNA-seq pipelines for DEG detection by choosing alignment tools that result in high ZeroMismatchPercentage. Although this preliminary study focused on two samples, nine different pipelines, and two metrics, we plan to include additional samples (i.e., SEQC



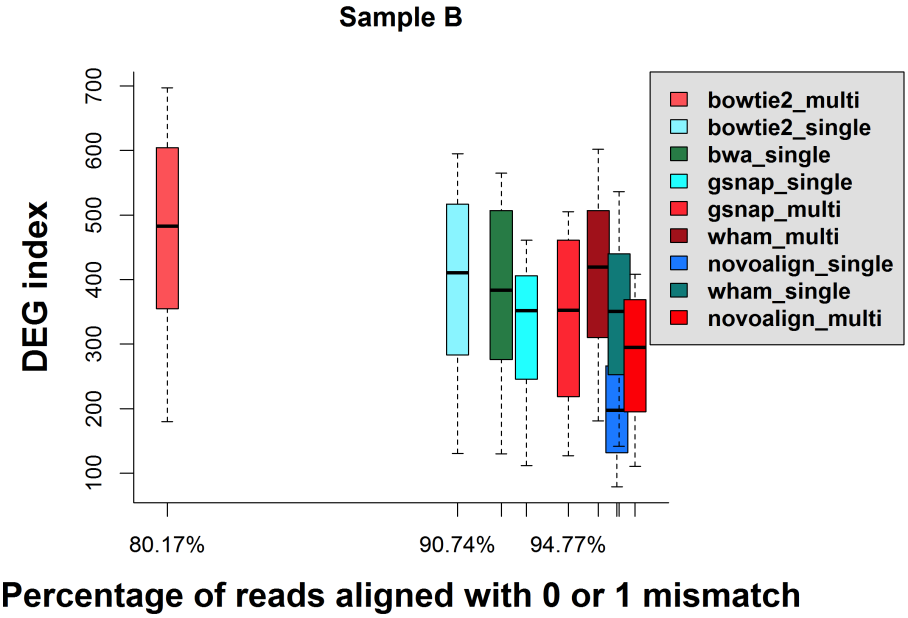


Figure A.7: The impact of alignment pipelines on gene expression estimation (Sample B).

samples C and D), pipelines, and metrics in a more comprehensive study.

APPENDIX B

THE IMPACT OF RNA-SEQ ALIGNERS ON GENE EXPRESSION ESTIMATION

Components of this chapter have been published as: Yang, C., Wu, P. Y., Tong, L., Phan, J., & Wang, M. (2015, September). The impact of RNA-seq aligners on gene expression estimation. In Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics (pp. 462-471). ACM. doi:10.1145/2808719.2808767

B.1 Abstract

While numerous RNA-seq data analysis pipelines are available, research has shown that the choice of pipeline influences the results of differentially expressed gene detection and gene expression estimation. Gene expression estimation is a key step in RNA-seq data analysis, since the accuracy of gene expression estimates profoundly affects the subsequent analysis. Generally, gene expression estimation involves sequence alignment and quantification, and accurate gene expression estimation requires accurate alignment. However, the impact of aligners on gene expression estimation remains unclear. We address this need by constructing nine pipelines consisting of nine spliced aligners and one quantifier. We then use simulated data to investigate the impact of aligners on gene expression estimation. To evaluate alignment, we introduce three alignment performance metrics, (1) the percentage of reads aligned, (2) the percentage of reads aligned with zero mismatch (ZeroMismatchPercentage), and (3) the percentage of reads aligned with at most one mismatch (ZeroOneMismatchPercentage). We then evaluate the impact of alignment performance on gene expression estimation using three metrics, (1) gene detection accuracy, (2) the number of genes falsely quantified (FalseExpNum), and (3) the number of genes with falsely estimated fold changes (FalseFcNum). We found that among various pipelines, FalseExpNum and FalseFcNum are correlated. Moreover, FalseExpNum is linearly correlated with

the percentage of reads aligned and ZeroMismatchPercentage, and FalseFcNum is linearly correlated with ZeroMismatchPercentage. Because of this correlation, the percentage of reads aligned and ZeroMismatchPercentage may be used to assess the performance of gene expression estimation for all RNA-seq datasets.

B.2 Introduction

RNA sequencing (i.e., RNA-seq) refers to the technologies and applications for high-throughput sequencing of RNA [3]. With the development of next-generation sequencing technology, RNA-seq has evolved to be a promising technology that plays an important role in several applications such as differential expression analysis, single nucleotide variation discovery, fusion gene detection, and co-expression network construction [25, 183, 184, 185, 186].

Typically, an RNA-seq data analysis pipeline includes (1) sequence read alignment, (2) expression quantification, (3) expression normalization, and (4) differentially expressed gene (DEG) detection. For each step of the pipeline, many algorithms or tools have been developed. Being aware of a large amount of combinations of RNA-seq data analysis pipelines, researchers have conducted comparative and quality control studies [181, 187, 44, 188, 189, 190, 176, 191] for quantifying the performance of tools or algorithms and ensuring the accuracy and reproducibility of RNA-seq. Conclusions from most studies support that the choice of pipelines affects the analysis results. For example, Grant et al. [176] evaluated various alignment algorithms and observed the discrepancy of alignment performance. Fonseca et al. [187] combined various alignment algorithms and three quantification tools to analyze the variance of detected and true gene expression levels, and proved that different analysis pipelines affected the gene expression levels. Soneson et al. [44] compared methods for differential expression analysis and found that shared differentially expressed genes detected by different methods varied significantly. Most of these studies focus on the comparison of algorithms or tools belonging to each step,

which cannot illustrate how the impact propagates through the steps of RNA-seq analysis pipelines. Although Fonseca et al. [187] combined aligners and quantifiers to investigate the variance of detected and true gene expression, they mainly compared the performance of the pipelines, and did not explain how alignment pipelines affected the gene expression estimates. The SEQC/MAQC-III consortium conducted a large-scale, multisite, cross-platform RNA-seq study that aimed to build standards for RNA-seq research from sample preparation to downstream analytics. They found that RNA-seq measurement performance depended on platforms and data analysis pipelines [181]. However, the choice of which pipeline researchers should apply still remains unclear. To solve this problem, the intuition is to conduct a pipeline-level comparative study for RNA-seq data analysis. However, the huge amount of pipelines impedes a comprehensive evaluation. Even though a comprehensive comparative study could be realized for some datasets, we cannot be assured of finding a pipeline that always outperforms other pipelines for all datasets. To ensure the accuracy and reproducibility of RNA-seq data analysis results, we need to investigate the cause of the performance variance among RNAseq data analysis pipelines. Indeed, if we can identify the impact of error propagation of the RNA-seq data analysis pipelines, we might be able to design the pipeline or redesign the tool or algorithms of each step to achieve better performance.

Gene expression quantification is a key step in the RNA-seq data analysis pipeline, and the accuracy of expression quantification can profoundly affect the subsequent analysis. However, accurate gene expression quantification requires accurate sequence read alignment. As previously mentioned, Fonseca et al. [187] evaluated the effect of different analysis pipelines on gene expression estimation and assessed the difference between true and estimated expression, but they mainly focused on the comparison of the pipelines and did not reveal why and how the choice of aligners and quantifiers influences the gene expression level. We investigate the impact of aligners on gene expression estimation and try to find indicators which can correlate the performance of aligners and gene expression

estimation.

B.3 Methods

The workflow of our study is shown in Figure B.1. To investigate the impact of RNA-seq aligners on gene expression estimation, we vary aligners which are specifically designed for genome alignment. For quantification tool, we use a fixed tool: HTSeq [30].

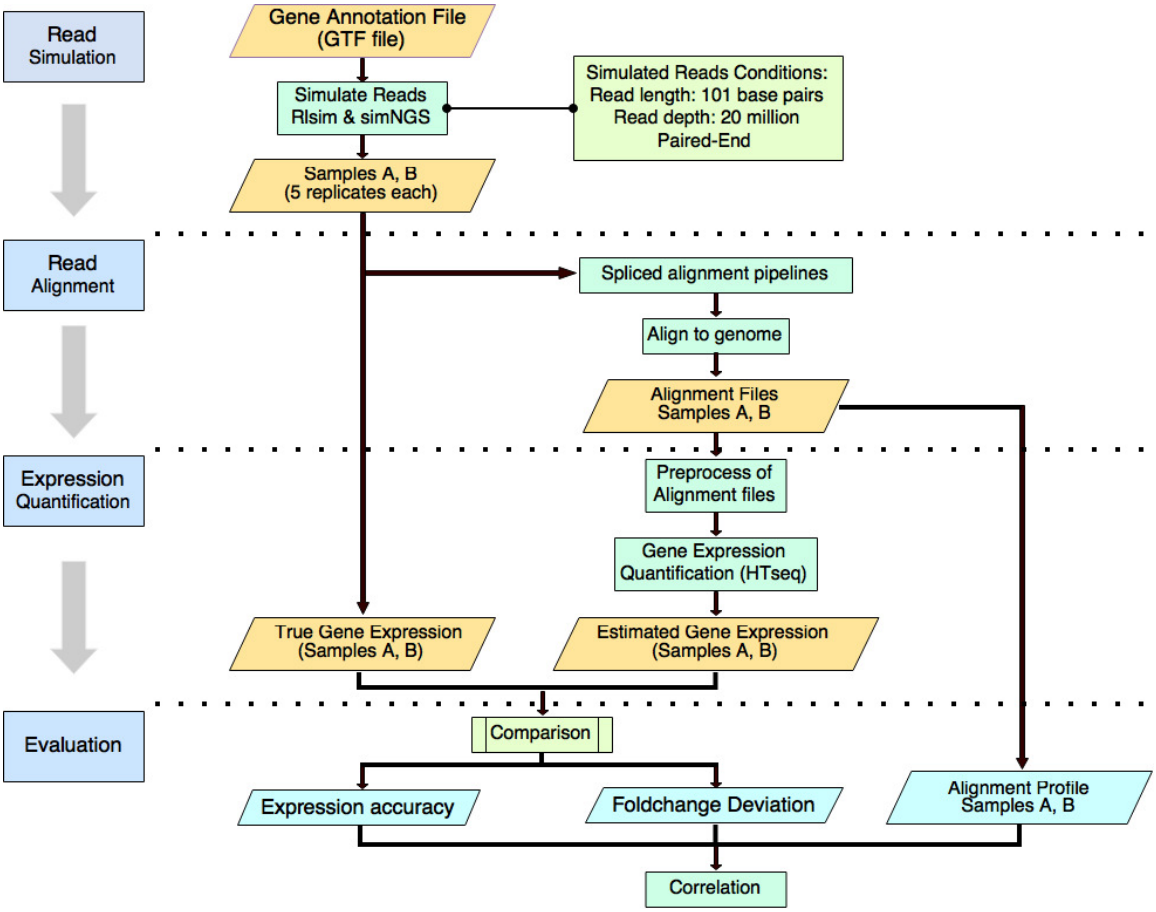


Figure B.1: The workflow of experimental design and data analysis.

B.3.1 Simulation of RNA-seq Dataset

Real RNA-seq datasets do not contain ground-truth information. To facilitate the investigation of the impact of RNA-seq aligners on gene expression estimation, we need to know the true expression level of every gene. Therefore, we use a simulated RNA-seq dataset for

this study. We employ rlsim [192] with simNGS [193] to generate RNA-seq data. rlsim integrates a collection of tools to simulate RNA-seq library construction [192] and can generate the simulated RNA fragments. simNGS can simulate observed reads from Illumina sequencing machines and incorporate noise due to sequencing. We apply rlsim to generate RNA fragments and simNGS to simulate RNA-seq reads.

For constructing the RNA library, we use the default setting of rlsim to generate 20 million RNA fragments based on the RefSeq gene annotation and the UCSC hg19 reference genome. First, we employ the "sel" tool from rlsim package to sample the expression level of each transcript from a mixture of gamma distributions including Gamma(5000, 0.1) and Gamma(10000, 100). Second, we adopt rlsim to generate RNA fragments from the previous FASTA file. With RNA fragments, we then employ simNGS to simulate paired-end reads: we use "s_6_4x.runfile", which is shipped with the simNGS package, to simulate 101bp paired-end reads from each fragment. Besides the absolute expression of each gene, we are also interested in the relative gene expression levels. Thus, we simulate two samples-Samples A and B-each of which has five replicates, and each replicate has 20 million paired-end reads. For gene expression fold changes, we follow the simulation strategy proposed by Zheng et al. [194]. Using the same simulated expression levels generated by "sel", we artificially introduce some differentially expressed genes with predefined fold changes. Sample A was simulated by using the original expression profile. For Sample B, we randomly choose 10% genes to be overexpressed, 10% genes underexpressed, and the rest 80% genes remain unchanged. Among all overexpressed and underexpressed genes, we randomly and equally assign a predefined fold change to each gene. Table B.1 summarizes the preset gene expression fold change. With these settings, we obtain two samples with built-in truths about absolute gene expression and relative expression fold changes.

Table B.1: Simulation strategy

Gene Types	Gene expression levels		Number of genes (25,678)[a]	Gene expression fold change A vs. B
	Sample A	Sample B		
I	Normal	Over expressed	576	1:2
			568	1:3
			579	1:4
			557	1:5
II	Normal	Under expressed	576	2:1
			589	3:1
			598	4:1
			519	5:1
III	Normal	Normal	21116	1:1

[a] Total number of genes.

B.3.2 Sequence Alignment

To analyze the impact of alignment on gene expression estimation, we use various alignment tools and a fixed quantification tool to control variables. Until now, researchers have developed many RNA-seq alignment tools or pipelines, which can be categorized as transcriptome aligners and genome aligners. Transcriptome aligners can reduce the alignment complexity by aligning sequence reads to known transcripts, while genome aligners directly align the reads to the genome and must address the reads derived from splice junctions [195]. However, transcriptome aligners are usually combined with isoform expression quantification, which need to be translated to gene expression levels if we are interested in the latter. Therefore, we select nine recently released spliced aligners, including Tophat2 [14], STAR [15], MapSplice [16], GSNAP_spliced [179], PASSION [19], OLego [196], Subread [197], SOAPSplice [198], and GEM [18]. We use UCSC hg19 as the reference genome. If the alignment tool supports multiple-hit mapping strategy, such as Tophat2, GSNAP_spliced, OLEGO, STAR, and Subread, we allow up to twenty hits for each read. For other options, we follow default settings.

B.3.3 Expression Quantification

For gene expression quantification, we use HTSeq (the intersection-nonempty mode) with RefSeq as the genome annotation. HTSeq is a count-based quantification tool, enabling us to compare the estimated gene expression to the built-in truth. Because counting multiple-hit reads (reads that have multiple mapping locations) might cause false positive differentially expressed genes [30], the default setting of HTSeq tends to discard all of these multiple-hit reads. However, discarding the multiple-hit reads may also incur false negative errors in terms of gene detection. For example, if one and only one mapping of a multiple-hit read is correct and we discard it, then the expression of the associated gene will be underestimated. In this study, we choose to keep all multiple-hit reads by removing the tag that HTSeq uses to identify the multiple-hit reads.

B.3.4 Performance Evaluation

Alignment Profile Construction

Here we propose to use the percentage of reads aligned (ReadsAlignedPercentage), the percentage of reads aligned with zero mismatch (ZeroMismatchPercentage) and at most one mismatch (ZeroOneMismatchPercentage) as the metrics for assessing alignment quality. We hypothesize that the percentage of reads aligned can quantify the mapping capability of an alignment pipeline, and reads aligned with less than one mismatch are more reliable for downstream expression estimates.

Gene Expression Evaluation

After quantifying the gene expression, we get the gene count number for each gene. Since we already know the true expression for each gene, we can compare the estimated gene expression to the built-in truth. However, not all reads can be aligned through alignment, and not all reads will be assigned to a specific gene (e.g., assigned to no-feature and am-

Table B.2: Definition of gene detection accuracy

		True expression	
	Total Gene #	**Expressed**	**Not Expressed**
Reconstructed expression	**Expressed**	TP	FP
	Not Expressed	FN	TN

biguous) during quantification, which indicates a portion of reads will be discarded and cannot account for the gene expression. If we directly compare estimated expression to true expression, discrepancy between them is definitely expected. To compensate for this discrepancy, we propose to use the following two metrics to measure expression accuracy: (i) the detection ability of genes (Table B.2) measured by Accuracy = (TP + TN) / (TP + TN + FP + FN) and (ii) the number of genes falsely quantified (Notation: FalseExpNum) measured by (Equation B.1), which normalizes the difference between gene counts by median gene expression in the ground truth and estimated expression respectively.

$$\text{FalseExpNum} = \sum_{i=1}^{n} I(|\frac{T_i - R_i}{T_i}|>\text{Threshold}) \tag{B.1}$$

where T_i and R_i represent the true and estimated expression level of the *i-th* gene after normalization, respectively; I is the indicator function (I = 1 if the formula in parentheses is true; I = 0 otherwise; we consider 0/0 = 0); Threshold is between 0.2 and 1; and n is the total number of genes. To determine a falsely quantified gene, we incoporate the threshold, which quantifies the deviations compared with true expression level. Generally, a larger threshold indicates more tolerance to the deviations.

Fold-change Variance Evaluation

Besides the absolute expression accuracy, we also evaluate the relative expression accuracy (fold changes). We compute gene expression fold changes between Samples A and B using estimated gene expression. We then compare the estimated fold changes to the ground truth.

We count the number of genes with falsely estimated fold changes (Notation: FalseFcNum) given by Equation B.2.

$$\text{FalseExpNum} = \sum_{i=1}^{n} I(|\frac{TFC_i - EFC_i}{TFC_i}| > \text{Threshold}) \tag{B.2}$$

where TFC_i and EFC_i means the true and estimated fold change of the *i-th* gene, respectively; I is the indicator function (I = 1 if the formula in parentheses is true; I = 0 otherwise; we consider 0/0 = 0); Threshold is between 0.2 and 1; and n is the total number of genes. Also, threshold is used to quantify the deviations of true fold change.

Correlation

Once we acquire the alignment profile (i.e., ReadsAlignedPercentage, ZeroMismatchPercentage, and ZeroOneMismatchPercentage) and the aforementioned evaluation metrics (i.e., the gene detection accuracy, FalseExpNum, and FalseFcNum) we apply linear regression analysis to model their relationship. Since the only difference among the gene expression estimation pipelines we use is the aligner, if any discrepancy exists in the gene expression, the only source would be aligner. Thus, the logic would be treating alignment profile as the explanatory variable, and the evaluation of gene expression as dependent variable. For real data, we do not know the built-in truth, and we can only compute the metrics for alignment profiles. If we can observe some correlation between alignment profile and expression evaluation, we might be able to predict the expression performance based on the alignment profile. Therefore, we fit linear regression between alignment profile and the evaluation metrics under various threshold values (to verify if the alignment profiles correlate with expression evaluation), and we compute adjusted R^2 value for each one.

B.4 Results and discussion

B.4.1 Alignment Profile

At the first sight, we might total the ratio of correctly aligned reads as the metric of alignment pipelines performance, since we know the true mapping location of each read. However, for real data, we are not aware of the true alignment of every read, which negates the feasibility to employ the alignment accuracy as the metric. Thus, we introduce alternative metrics. The first metric in the alignment profile is the percentage of reads aligned. For every aligner, we observed that the percentage of reads aligned (ReadsAlignedPercentage) were almost of the same value in both Samples A and B, therefore we plotted them in one figure (Figure B.2). The small error bars indicate consistent performance among both Samples A and B. Except GEM, the ReadsAlignedPercentage of most aligners were over 90%.

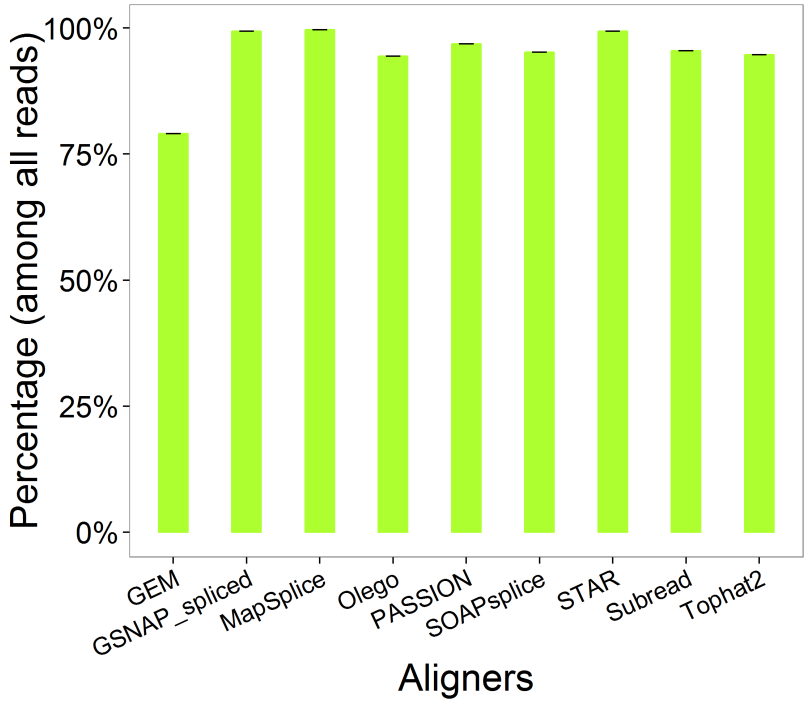


Figure B.2: The percentage of reads aligned.

Then, with the alignment results, we computed the percentage of reads aligned with

Table B.3: The rank of alignment profiles

Aligner	ReadsAligned Percentage	ZeroMismatch Percentage	OneMismatch Percentage
GEM	9	9	9
GSNAP_spliced	3	2	2
MapSplice	1	6	7
Olego	8	5	6
PASSION	4	1	5
SOAPsplice	6	7	8
STAR	2	4	3
Subread	5	8	4
Tophat2	7	3	1

zero or one mismatch. For each aligner, we found that both ZeroMismatchPercentage and ZeroOneMismatchPercentage were almost the same in Samples A and B. As we can see from Figure B.3 (each column includes all the replicates of Samples A and B), the reads aligned with zero and one mismatch can account for the majority of the aligned reads (over 80%). In addition, we used ANOVA to analyze the difference among the metrics of all the aligners (we apply ANOVA to any two aligners metrics). And we observed that ReadsAlignedPercentage, ZeroMismatchPercentage and ZeroOneMismatchPercentage are all significantly different among different aligners since all the p-values are less than 0.001. We also ranked the aligners according to the above three metrics separately (Table B.3).

B.4.2 Expression and Fold-change Evaluation

Figure B.4 displays the gene detection accuracy of each pipeline. For most pipelines, gene detection accuracy is at the same level (up to 90%), and show little difference, indicating that the gene detection accuracy might not be an appropriate metric for the evaluation of gene expression.

For the number of genes falsely quantified (FalseExpNum), we observed significant discrepancy among pipelines (Figure B.5). With the threshold increases, FalseExpNum

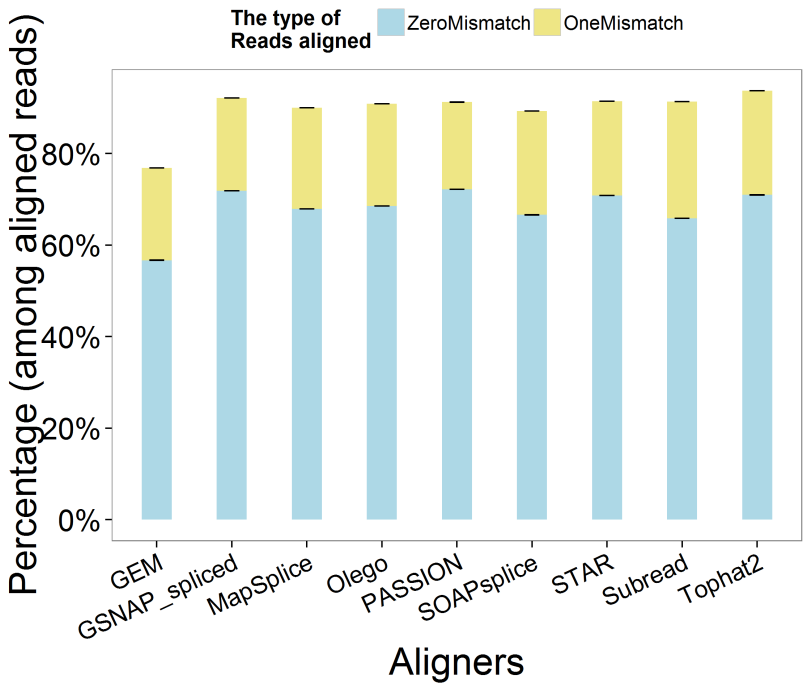


Figure B.3: The percentage of reads aligned with 0 or 1 mismatch.

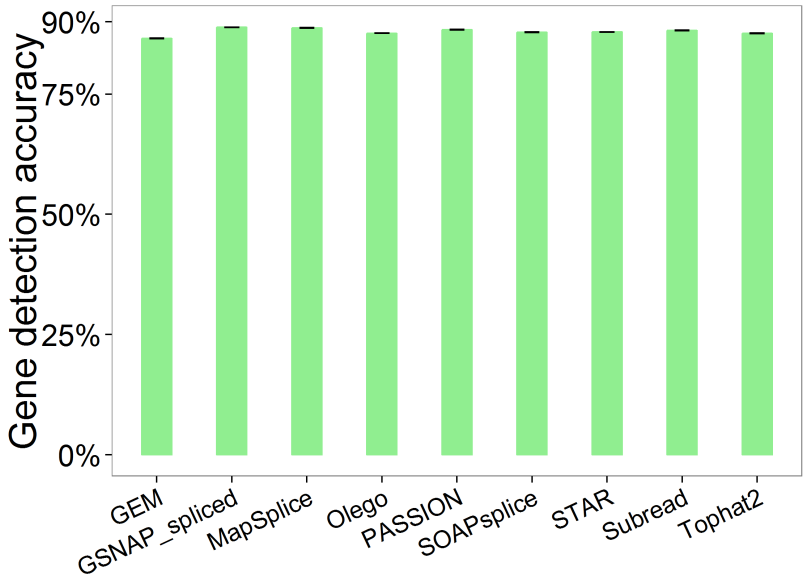


Figure B.4: The accuracy of gene detection.

decreases. This is reasonable because a larger threshold means higher tolerance to false quantified genes, which results in less number of genes falsely quantified. From Figure B.6, we can observe that the number of genes with falsely estimated fold changes (FalseFcNum)

also varies among pipelines and shows a similar trend with increase of threshold.

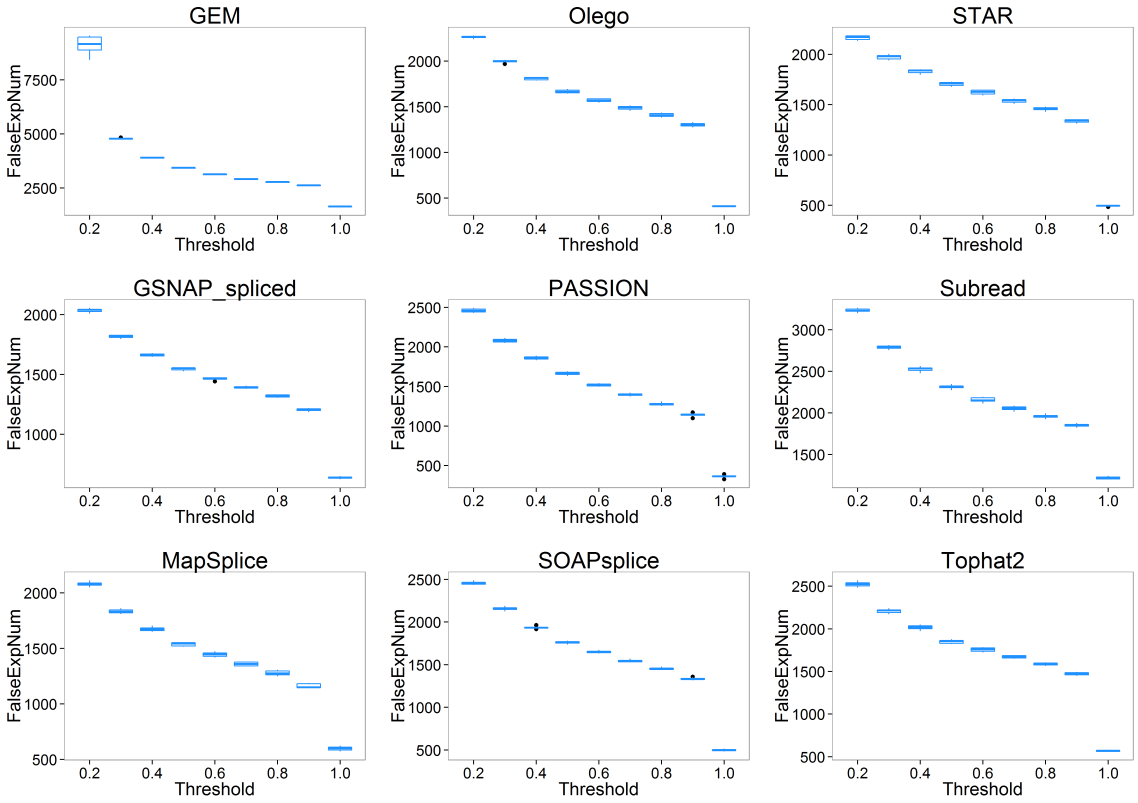


Figure B.5: The number of genes falsely quantified.

Logically, only the genes falsely quantified might have false estimated fold change. To investigate the consistency between the two metrics (FalseExpNum and FalseFcNum), we computed the Pearson correlation coefficient (Table B.4). As we can see in Table B.4, FalseExpNum and FalseFcNum show significant linear correlation with each other, suggesting that both FalseExpNum and FalseFcNum can be equally employed as the metric of gene expression estimates. However, comparing FalseExpNum and FalseFcNum, we observed that FalseExpNum was generally larger than FalseFcNum, indicating that even though some genes have been falsely quantified, the fold changes of these genes will not be affected.

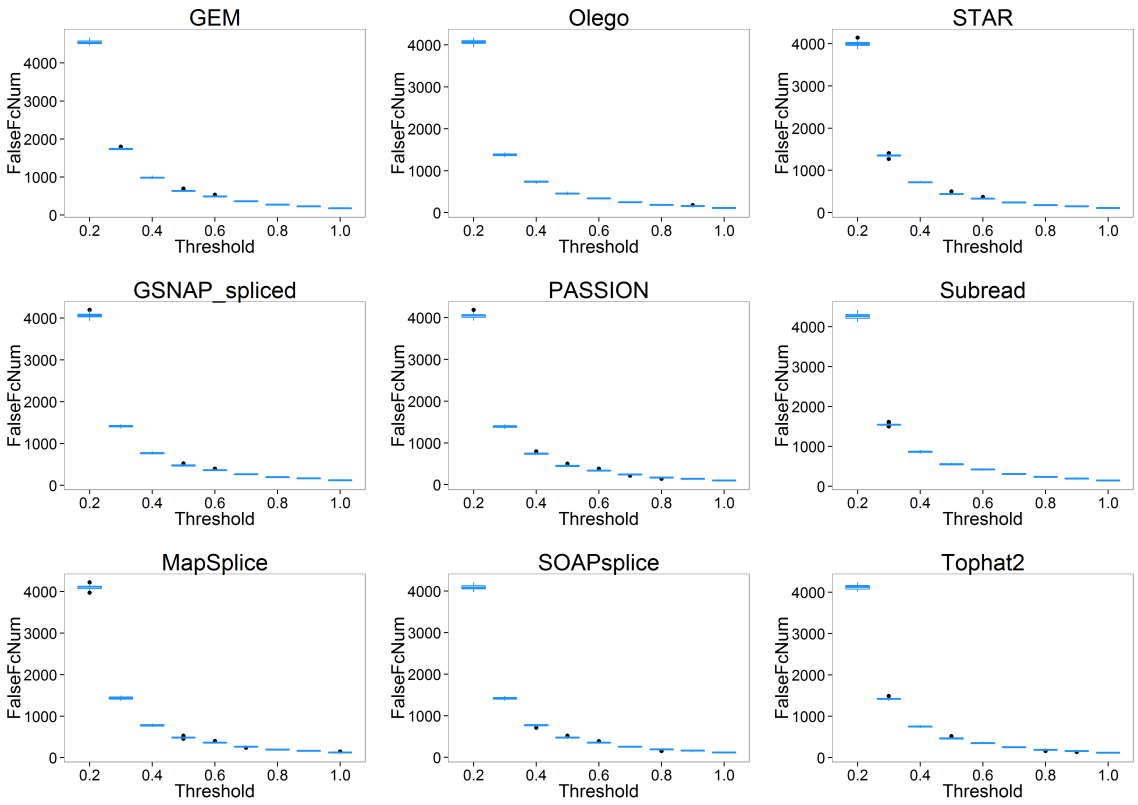


Figure B.6: The number of genes with falsely estimated fold-change.

B.4.3 Correlation

Obtaining the alignment profile (ReadsAlignedPercentage, ZeroMismatchPercentage and ZeroOneMismatchPercentage) and the expression evaluation (FalseExpNum and FalseFcNum), we applied linear regression to fit their relationship. Since three metrics of alignment profile were available, we fitted the model (1) with only ReadsAlignedPercentage, (2) with ZeroMismatchPercentage, (3) with ZeroOneMismatchPercentage, (4) with both ReadsAlignedPercentage and ZeroMismatchPercentage, and (5) with both ReadsAlignedPercentage and ZeroOneMismatchPercentage. From Table B.5, for FalseExpNum, we can see among all the linear regressions, when fitting with both ReadsAlignedPercentage and ZeroMismatchPercentage, the adjusted R^2 is generally larger than others. In contrast, for FalseFcNum (Table B.6), we found that when fitting with ZeroMismatchPercentage, the adjusted R^2 is larger.

Table B.4: Correlation efficient of FalseExpNum and FalseFcNum

Variance threshold	r^2	P value
0.2	0.8879	0.0001
0.3	0.9173	0.0000
0.4	0.8945	0.0001
0.5	0.8949	0.0001
0.6	0.8864	0.0002
0.7	0.8938	0.0001
0.8	0.8906	0.0001
0.9	0.8956	0.0001
1	0.9613	0.0000

Overall, combined with Tables B.5 and B.6, Figures B.7 and B.8 show the key findings of our study. FalseExpNum shows linear correlation with ReadsAlignedPercentage and ZeroMismatchPercentage, and FalseFcNum shows linear correlation with ZeroMismatchPercentage. Since FalseExpNum and FalseFcNum are the metrics of gene expression estimation, and ReadsAlignedPercentage and ZeroMismatchPercentage are metrics of alignment, the linear correlation implies that with the increase of ReadsAlignedPercentage and ZeroMismatchPercentage, the performance of gene expression estimation will improve. We believe our foremost hypothesis might help to explain this phenomenon: reads aligned with zero mismatch have higher probability to be correctly mapped, and ReadsAlignedPercentage quantifies the portion of reads that have been mapped. Combining ReadsAlignedPercentage and ZeroMismatchPercentage, we might assess the performance of alignment, while the better the performance of alignment, the better gene expression estimates. Our finding also suggests applying aligners which can produce higher ReadsAlignedPercentage and ZeroMismatchPercentage when conducting gene expression estimates-related analysis, such as DEG detection.

Table B.5: Linear regression of FalseExpNum

FalseExpNum	~ReadsAligned Percentage		~ZeroMismatch Percentage		~ZeroOneMismatchPercentage		~ReadsAligned Percentage + ZeroMismatch Percentage		~ReadsAligned Percentage + ZeroOneMismatchPercentage	
Threshold	R^2(adj)	p-value	R^2(adj)	p-value	R^2(adj)	p-value	R^2(adj)	p-value	R^2(adj)	p-value
0.2	0.9140	0.0000	0.7978	0.0007	0.8922	0.0001	0.9187	0.0001	**0.9502**	0.0000
0.3	0.9010	0.0001	0.8172	0.0005	0.8086	0.0006	**0.9148**	0.0001	0.9024	0.0001
0.4	0.8762	0.0001	0.8075	0.0006	0.7612	0.0013	**0.8909**	0.0002	0.8656	0.0003
0.5	0.8597	0.0002	0.8043	0.0006	0.7386	0.0018	**0.8767**	0.0003	0.8441	0.0005
0.6	0.8492	0.0003	0.8004	0.0007	0.7172	0.0024	**0.8670**	0.0004	0.8292	0.0007
0.7	0.8327	0.0004	0.7927	0.0008	0.6906	0.0034	**0.8510**	0.0005	0.8077	0.0010
0.8	0.8200	0.0005	0.7909	0.0008	0.6736	0.0041	**0.8412**	0.0006	0.7920	0.0013
0.9	0.8101	0.0006	0.7894	0.0008	0.6572	0.0049	**0.8336**	0.0007	0.7796	0.0015
1	0.5562	0.0128	0.7169	0.0025	0.5394	0.0147	**0.6709**	0.0084	0.5144	0.0192

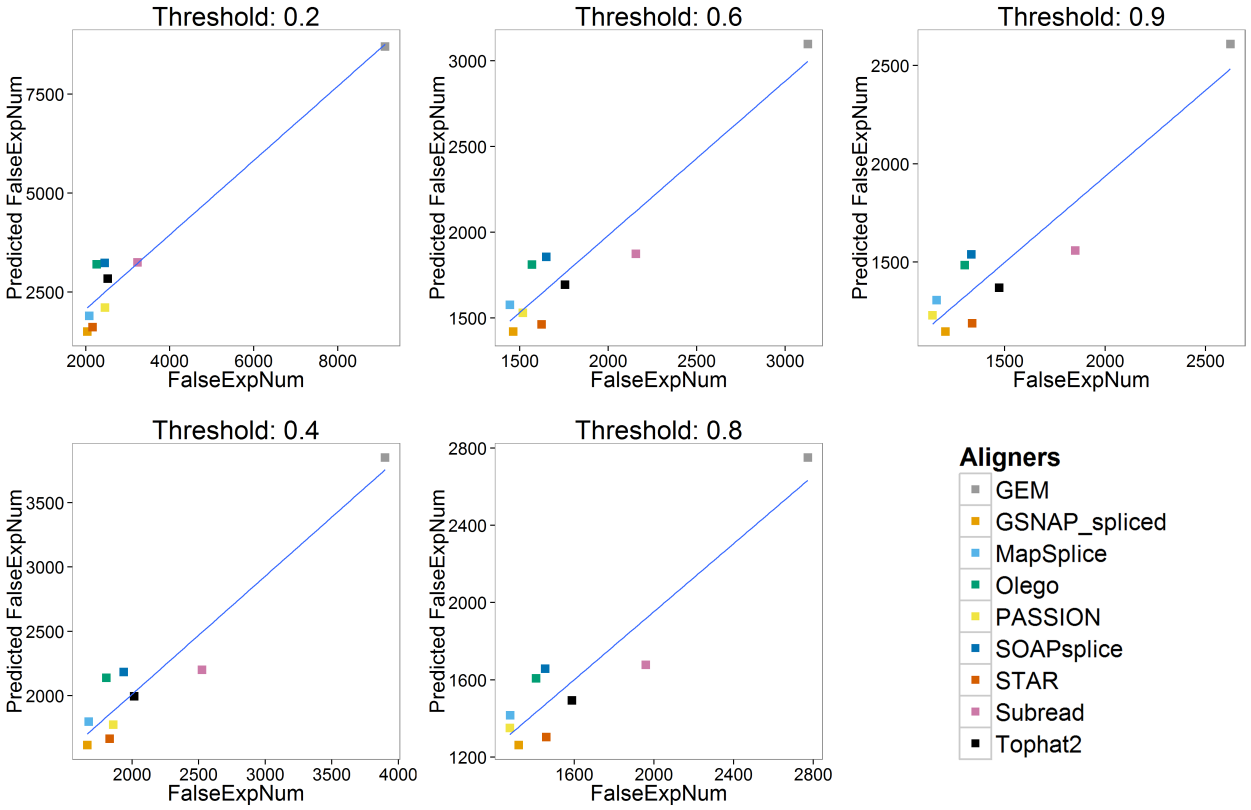


Figure B.7: Correlation between predicted FalseExpNum (with ReadsAlignedPercentage and ZeroMismatchPercentage) and true FalseExpNum.

Table B.6: Linear regression of FalseFcNum

FalseFcNum	~ReadsAligned Percentage		~ZeroMismatch Percentage		~ZeroOneMismatchPercentage		~ReadsAligned Percentage + ZeroMismatch Percentage		~ReadsAligned Percentage + ZeroOneMismatchPercentage	
Threshold	R^2(adj)	p-value	R^2(adj)	p-value	R^2(adj)	p-value	R^2(adj)	p-value	R^2(adj)	p-value
0.2	0.7897	0.0008	0.8361	0.0003	0.6997	0.0030	**0.8520**	0.0006	0.7682	0.0018
0.3	0.7378	0.0019	**0.8365**	0.0003	0.7172	0.0024	0.8300	0.0009	0.7349	0.0029
0.4	0.6859	0.0036	**0.8338**	0.0004	0.7086	0.0027	0.8128	0.0014	0.6980	0.0047
0.5	0.6756	0.0040	**0.8389**	0.0003	0.6945	0.0032	0.8160	0.0014	0.6827	0.0054
0.6	0.6664	0.0045	**0.8281**	0.0004	0.6933	0.0033	0.8034	0.0017	0.6770	0.0058
0.7	0.6708	0.0042	**0.8383**	0.0003	0.6872	0.0035	0.8147	0.0015	0.6752	0.0058
0.8	0.6524	0.0052	**0.8489**	0.0003	0.6643	0.0046	0.8243	0.0013	0.6496	0.0072
0.9	0.6600	0.0048	**0.8501**	0.0003	0.6772	0.0040	0.8261	0.0013	0.6625	0.0065
1	0.6628	0.0046	**0.8535**	0.0002	0.6718	0.0042	0.8301	0.0012	0.6605	0.0065

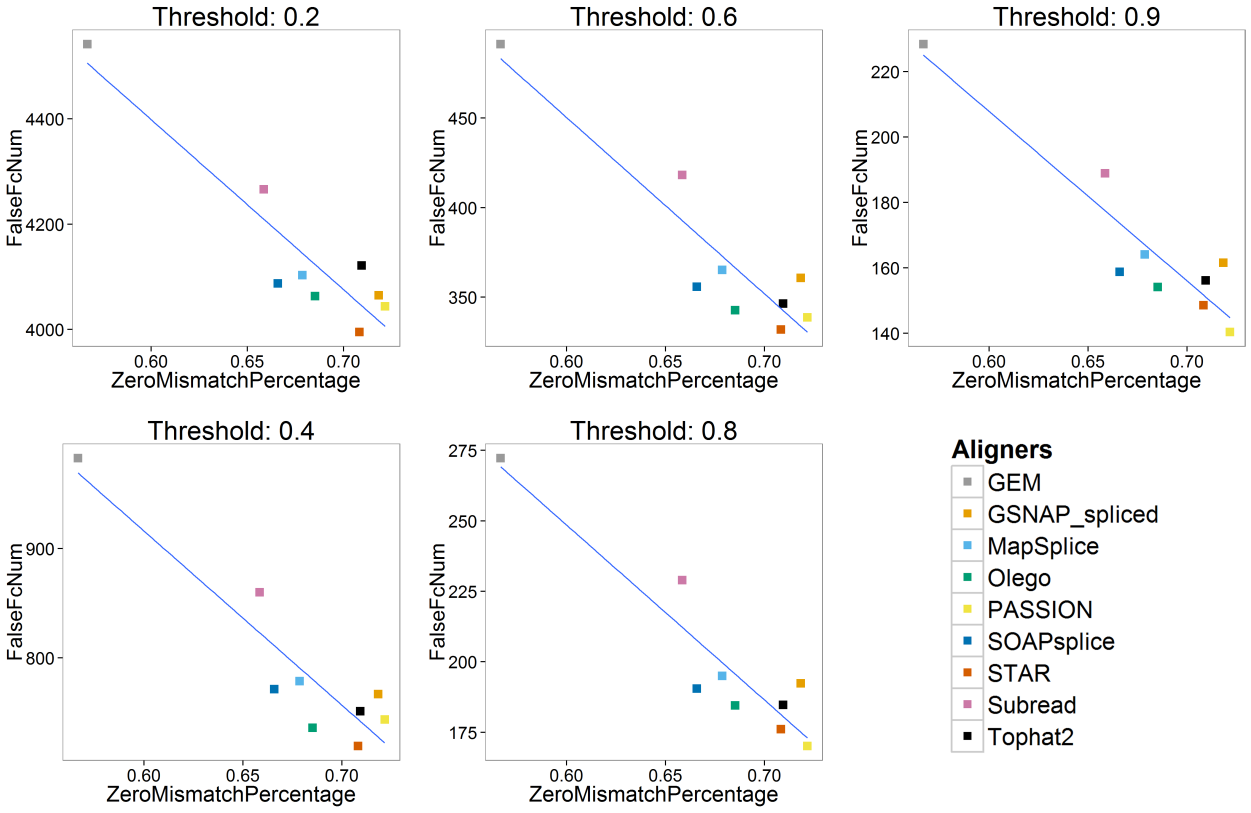


Figure B.8: Correlation between FalseFcNum and ZeroMismatchPercentage.

B.5 Conclusions

We analyzed the impact of RNA-seq aligners on gene expression estimation by constructing RNA-seq data analysis pipelines with nine different aligners and one quantification tool, HTSeq. Using simulated RNA-seq data, we have the true gene expression and true gene fold change between samples.

We profiled the alignment performance with (1) the percentage of reads aligned (ReadsAlignedPercentage), (2) the percentage of reads aligned with zero mismatch (ZeroMismatchPercentage), and the percentage of reads aligned with at most one mismatch (ZeroOneMismatchPercentage). We observed that for most aligners, the ReadsAlignedPercentage can be over 90%, and the reads aligned with zero or one mismatch can account for over 80% of aligned reads.

We evaluated the gene expression estimation with three metrics: (1) the accuracy of gene detection, (2) the number of genes falsely quantified (FalseExpNum), and (3) the number of genes with falsely estimated fold change (FalseFcNum). We found that for most pipelines, the accuracy of gene detection shows few discrepancies suggesting gene detection accuracy might not be a suitable metric for gene expression estimation. In contrast, for FalseExpNum and FalseFcNum, the discrepancy among pipelines is more significant. In addition, we observed linear correlation between FalseExpNum and FalseFcNum, suggesting both FalseExpNum and FalseFcNum might be equally applied as the metric of gene expression estimation. However, FalseExpNum is generally larger than FalseFcNum, implying that the fold change of some genes will not be affected even though they are falsely quantified.

We applied linear regression to model the relationship between the alignment profile (ReadsAlignedPercentage, ZeroMismatchPercentage and ZeroOneMismatchPercentage) and the evaluation of gene expression (FalseExpNum and FalseFcNum). We observed that FalseExpNum shows linear correlation with ReadsAlignedPercentage and ZeroMis-

matchPercentage, and FalseFcNum shows linear correlation with ZeroMismatchPercentage. An explanation might be: (1) the reads aligned with zero mismatch are more likely to be correctly mapped, which contributes more to accurate quantification; (2) the percentage of reads aligned represents the amount of reads that might be correctly mapped. Therefore, ZeroMismatchPercentage and ReadsAlignedPercentage might be combined as indicators of the performance of gene expression estimates. We plan to verify this by applying our method to real data in a future study. Since ZeroMismatchPercentage and ReadsAlignedPercentage can be calculated without knowing the true alignment, indicating we can calculate these two metrics for real data. Once got the above two alignment metrics, we might assess the performance of gene expression estimation.

Overall, based on the results of our experiment, when conducting gene expression estimation, we suggest applying aligners that produce higher ReadsAlignedPercentage and ZeroMismatchPercentage. Using this criterion, STAR, PASSION and GSNAP_spliced aligners outperform other aligners when applied to our simulated dataset (Table B.3).

APPENDIX C

SELECTED PUBLICATIONS

1. **Yang, C.**, Yang, L. S., Zhou, M., Zhang C. J., Xie, H. L., Wang, M. D., & Zhu, H. Q. An ab initio lncRNA identification and functional annotation tool based on deep learning. (Submitted to Bioinformatics).

2. **Yang, C.**, & Zhu, H. Q. (2016, October). A lncRNA prediction tool based on deep learning algorithm. The 7th National Conference on Bioinformatics and System Biology.

3. Tong, L., **Yang, C.**, Wu, P. Y., & Wang, M. D. (2016, February). Evaluating the impact of sequencing error correction for RNA-seq data with ERCC RNA spike-in controls. In Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on (pp. 74-77). IEEE. (EI Index)

4. **Yang, C.**, Wu, P. Y., Tong, L., Phan, J., & Wang, M. D. (2015, September). The impact of RNA-seq aligners on gene expression estimation. In Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics (pp. 462-471). ACM. (EI Index)

5. **Yang, C.**, Wu, P. Y., Phan, J. H., & Wang, M. D. (2014, December). The impact of RNA-seq alignment pipeline on detection of differentially expressed genes. In Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on (pp. 1376-1379). IEEE. (EI Index)

APPENDIX D

COPYRIGHT PERMISSIONS

Title: The impact of RNA-seq alignment pipeline on detection of differentially expressed genes

Conference Proceedings: Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on

Author: Cheng Yang

Publisher: IEEE

Date: Dec. 2014

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Nov 06, 2017

This Agreement between Cheng Yang ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") consists of your license details and the terms and conditions provided by Association for Computing Machinery, Inc. and Copyright Clearance Center.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	4223391150556
License date	Nov 06, 2017
Licensed Content Publisher	Association for Computing Machinery, Inc.
Licensed Content Publication	Proceedings
Licensed Content Title	The impact of RNA-seq aligners on gene expression estimation
Licensed Content Author	Cheng Yang, et al
Licensed Content Date	Sep 9, 2015
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	No
Format	Electronic
Portion	Full article
Will you be translating?	No
Order reference number	
Title of your thesis/dissertation	Cheng Yang's Dissertation
Expected completion date	Jan 2018
Estimated size (pages)	150
Requestor Location	Cheng Yang Yiheyuan Road 5 Beijing, 100871 China Attn: Cheng Yang
Billing Type	Credit Card
Credit card info	American Express ending in 1006
Credit card expiration	03/2020
Total	**8.00 USD**
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

1. The publisher of this copyrighted material is Association for Computing Machinery, Inc. (ACM). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at).
2. ACM reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.
3. ACM hereby grants to licensee a non-exclusive license to use or republish this ACM-copyrighted material* in secondary works (especially for commercial distribution) with the stipulation that consent of the lead author has been obtained independently. Unless otherwise stipulated in a license, grants are for one-time use in a single edition of the work, only with a maximum distribution equal to the number that you identified in the licensing process. Any additional form of republication must be specified according to the terms included at the time of licensing.

*Please note that ACM cannot grant republication or distribution licenses for embedded third-party material. You must confirm the ownership of figures, drawings and artwork prior to use.
4. Any form of republication or redistribution must be used within 180 days from the date stated on the license and any electronic posting is limited to a period of six months unless an extended term is selected during the licensing process. Separate subsidiary and subsequent republication licenses must be purchased to redistribute copyrighted material on an extranet. These licenses may be exercised anywhere in the world.
5. Licensee may not alter or modify the material in any manner (except that you may use, within the scope of the license granted, one or more excerpts from the copyrighted material, provided that the process of excerpting does not alter the meaning of the material or in any way reflect negatively on the publisher or any writer of the material).
6. Licensee must include the following copyright and permission notice in connection with any reproduction of the licensed material: "[Citation] © YEAR Association for Computing Machinery, Inc. Reprinted by permission." Include the article DOI as a link to the definitive version in the ACM Digital Library. Example: Charles, L. "How to Improve Digital Rights Management," Communications of the ACM, Vol. 51:12, © 2008 ACM, Inc. http://doi.acm.org/10.1145/nnnnnn.nnnnnn (where nnnnnn.nnnnnn is replaced by the actual number).
7. Translation of the material in any language requires an explicit license identified during the licensing process. Due to the error-prone nature of language translations, Licensee must include the following copyright and permission notice and disclaimer in connection with any reproduction of the licensed material in translation: "This translation is a derivative of ACM-copyrighted material. ACM did not prepare this translation and does not guarantee that it is an accurate copy of the originally published work. The original intellectual property contained in this work remains the property of ACM."
8. You may exercise the rights licensed immediately upon issuance of the license at the end of the licensing transaction, provided that you have disclosed complete and accurate details of your proposed use. No license is finally effective unless and until full payment is received from you (either by CCC or ACM) as provided in CCC's Billing and Payment terms and conditions.
9. If full payment is not received within 90 days from the grant of license transaction, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted.
10. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.
11. ACM makes no representations or warranties with respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.
12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.
13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.
14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).
15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.
16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.
17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com
18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.
Special Terms:

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

REFERENCES

[1] G. Piétu, R. Mariage-Samson, N.-A. Fayein, C. Matingou, E. Eveno, R. Houlgatte, C. Decraene, Y. Vandenbrouck, F. Tahi, M.-D. Devignes, *et al.*, "The genexpress image knowledge base of the human brain transcriptome: A prototype integrated resource for functional and computational genomics," *Genome Research*, vol. 9, no. 2, pp. 195–209, 1999.

[2] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by rna-seq," *Nature Methods*, vol. 5, no. 7, pp. 621–628, 2008.

[3] Z. Wang, M. Gerstein, and M. Snyder, "Rna-seq: A revolutionary tool for transcriptomics," *Nature Reviews. Genetics*, vol. 10, no. 1, pp. 57–63, 2009.

[4] M. Garber, M. G. Grabherr, M. Guttman, and C. Trapnell, "Computational methods for transcriptome annotation and quantification using rna-seq," *Nature Methods*, vol. 8, no. 6, pp. 469–477, 2011.

[5] V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler, "Serial analysis of gene expression," *Science*, vol. 270, no. 5235, p. 484, 1995.

[6] T. Shiraki, S. Kondo, S. Katayama, K. Waki, T. Kasukawa, H. Kawaji, R. Kodzius, A. Watahiki, M. Nakamura, T. Arakawa, *et al.*, "Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage," *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15 776–15 781, 2003.

[7] R. Lowe, N. Shirley, M. Bleackley, S. Dolan, and T. Shafee, "Transcriptomics technologies," *PLoS computational biology*, vol. 13, no. 5, e1005457, 2017.

[8] I. Barbulovic-Nad, M. Lucente, Y. Sun, M. Zhang, A. R. Wheeler, and M. Bussmann, "Bio-microarray fabrication techniquesa review," *Critical reviews in biotechnology*, vol. 26, no. 4, pp. 237–259, 2006.

[9] E. R. Mardis, "The impact of next-generation sequencing technology on genetics," *Trends in genetics*, vol. 24, no. 3, pp. 133–141, 2008.

[10] S. Parekh, C. Ziegenhain, B. Vieth, W. Enard, and I. Hellmann, "The impact of amplification on differential expression analyses by rna-seq," *Scientific reports*, vol. 6, 2016.

[11] S. Zhao, W.-P. Fung-Leung, A. Bittner, K. Ngo, and X. Liu, "Comparison of rna-seq and microarray in transcriptome profiling of activated t cells," *PLOS ONE*, vol. 9, no. 1, e78644, 2014.

[12] Y. Sui, X. Zhao, T. P. Speed, and Z. Wu, "Background adjustment for dna microarrays using a database of microarray experiments," *Journal of Computational Biology*, vol. 16, no. 11, pp. 1501–1515, 2009.

[13] K. J. Mantione, R. M. Kream, H. Kuzelova, R. Ptacek, J. Raboch, J. M. Samuel, and G. B. Stefano, "Comparing bioinformatic gene expression profiling methods: Microarray and rna-seq," *Medical science monitor basic research*, vol. 20, p. 138, 2014.

[14] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "Tophat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome Biology*, vol. 14, no. 4, R36, 2013.

[15] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras, "Star: Ultrafast universal rna-seq aligner," *Bioinformatics*, vol. 29, no. 1, pp. 15–21, 2013.

[16] K. Wang, D. Singh, Z. Zeng, S. J. Coleman, Y. Huang, G. L. Savich, X. He, P. Mieczkowski, S. A. Grimm, C. M. Perou, *et al.*, "Mapsplice: Accurate mapping of rna-seq reads for splice junction discovery," *Nucleic Acids Research*, gkq622, 2010.

[17] D. Kim, B. Langmead, and S. L. Salzberg, "Hisat: A fast spliced aligner with low memory requirements," *Nature methods*, vol. 12, no. 4, pp. 357–360, 2015.

[18] S. Marco-Sola, M. Sammeth, R. Guigó, and P. Ribeca, "The gem mapper: Fast, accurate and versatile alignment by filtration," *Nature Methods*, vol. 9, no. 12, pp. 1185–1188, 2012.

[19] Y. Zhang, E.-W. Lameijer, P. AC't Hoen, Z. Ning, P. E. Slagboom, and K. Ye, "Passion: A pattern growth algorithm-based pipeline for splice junction detection in paired-end rna-seq data," *Bioinformatics*, vol. 28, no. 4, pp. 479–486, 2012.

[20] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," *Nature Methods*, vol. 9, no. 4, pp. 357–359, 2012.

[21] H. Li and R. Durbin, "Fast and accurate long-read alignment with burrows–wheeler transform," *Bioinformatics*, vol. 26, no. 5, pp. 589–595, 2010.

[22] Y. Li, J. M. Patel, and A. Terrell, "Wham: A high-throughput sequence alignment method," *ACM Transactions on Database Systems (TODS)*, vol. 37, no. 4, p. 28, 2012.

[23] B. Li, V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey, "Rna-seq gene expression estimation with read mapping uncertainty," *Bioinformatics*, vol. 26, no. 4, pp. 493–500, 2009.

[24] B. J. Haas, A. Papanicolaou, M. Yassour, M. Grabherr, P. D. Blood, J. Bowden, M. B. Couger, D. Eccles, B. Li, M. Lieber, *et al.*, "De novo transcript sequence reconstruction from rna-seq: Reference generation and analysis with trinity," *Nature protocols*, vol. 8, no. 8, 2013.

[25] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter, "Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks," *Nature Protocols*, vol. 7, no. 3, pp. 562–578, 2012.

[26] M. Pertea, D. Kim, G. M. Pertea, J. T. Leek, and S. L. Salzberg, "Transcript-level expression analysis of rna-seq experiments with hisat, stringtie and ballgown.," *Nature protocols*, vol. 11, no. 9, pp. 1650–1667, 2016.

[27] G. Robertson, J. Schein, R. Chiu, R. Corbett, M. Field, S. D. Jackman, K. Mungall, S. Lee, H. M. Okada, J. Q. Qian, *et al.*, "De novo assembly and analysis of rna-seq data," *Nature methods*, vol. 7, no. 11, pp. 909–912, 2010.

[28] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, *et al.*, "Trinity: Reconstructing a full-length transcriptome without a genome from rna-seq data," *Nature biotechnology*, vol. 29, no. 7, p. 644, 2011.

[29] T. Steijger, J. F. Abril, P. G. Engström, F. Kokocinski, T. J. Hubbard, R. Guigó, J. Harrow, P. Bertone, R. Consortium, *et al.*, "Assessment of transcript reconstruction methods for rna-seq," *Nature Methods*, vol. 10, no. 12, pp. 1177–1184, 2013.

[30] S. Anders, P. T. Pyl, and W. Huber, "Htseq–a python framework to work with high-throughput sequencing data," *Bioinformatics*, btu638, 2014.

[31] B. Li and C. N. Dewey, "Rsem: Accurate transcript quantification from rna-seq data with or without a reference genome," *BMC Bioinformatics*, vol. 12, no. 1, p. 323, 2011.

[32] M. Pertea, G. M. Pertea, C. M. Antonescu, T.-C. Chang, J. T. Mendell, and S. L. Salzberg, "Stringtie enables improved reconstruction of a transcriptome from rna-seq reads," *Nature biotechnology*, vol. 33, no. 3, pp. 290–295, 2015.

[33] A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, M. W. Szcześniak, D. J. Gaffney, L. L. Elo, X. Zhang, *et al.*, "A survey of best practices for rna-seq data analysis," *Genome biology*, vol. 17, no. 1, p. 13, 2016.

[34] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, *et al.*, "A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis," *Briefings in bioinformatics*, vol. 14, no. 6, pp. 671–683, 2013.

[35] Y. Lin, K. Golovnina, Z.-X. Chen, H. N. Lee, Y. L. S. Negron, H. Sultana, B. Oliver, and S. T. Harbison, "Comparison of normalization and differential expression analyses using rna-seq data from 726 individual drosophila melanogaster," *BMC genomics*, vol. 17, no. 1, p. 28, 2016.

[36] R. Patro, S. M. Mount, and C. Kingsford, "Sailfish enables alignment-free isoform quantification from rna-seq reads using lightweight algorithms," *Nature biotechnology*, vol. 32, no. 5, pp. 462–464, 2014.

[37] Z. Zhang and W. Wang, "Rna-skim: A rapid method for rna-seq quantification at transcript level," *Bioinformatics*, vol. 30, no. 12, pp. i283–i292, 2014.

[38] R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford, "Salmon provides fast and bias-aware quantification of transcript expression," *Nature Methods*, vol. 14, no. 4, pp. 417–419, 2017.

[39] N. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal rna-seq quantification," *arXiv preprint arXiv:1505.02710*, 2015.

[40] L. Wang, Z. Feng, X. Wang, X. Wang, and X. Zhang, "Degseq: An r package for identifying differentially expressed genes from rna-seq data," *Bioinformatics*, vol. 26, no. 1, pp. 136–138, 2009.

[41] S. Anders, "Analysing rna-seq data with the deseq package," *Molecular Biology*, vol. 43, no. 4, pp. 1–17, 2010.

[42] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "Edger: A bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2010.

[43] A. C. Frazee, G. Pertea, A. E. Jaffe, B. Langmead, S. L. Salzberg, and J. T. Leek, "Ballgown bridges the gap between transcriptome assembly and expression analysis," *Nature biotechnology*, vol. 33, no. 3, pp. 243–246, 2015.

[44] C. Soneson and M. Delorenzi, "A comparison of methods for differential expression analysis of rna-seq data," *BMC Bioinformatics*, vol. 14, no. 1, p. 91, 2013.

[45] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of rna-seq data," *Genome Biology*, vol. 11, no. 3, R25, 2010.

[46] A. F. Palazzo and T. R. Gregory, "The case for junk dna," *PLoS genetics*, vol. 10, no. 5, e1004351, 2014.

[47] A. Fatica and I. Bozzoni, "Long non-coding rnas: New players in cell differentiation and development," *Nature Reviews. Genetics*, vol. 15, pp. 7–21, 2014.

[48] T. Derrien, R. Johnson, G. Bussotti, A. Tanzer, S. Djebali, H. Tilgner, G. Guernec, D. Martin, A. Merkel, D. G. Knowles, J. Lagarde, L. Veeravalli, X. Ruan, Y. Ruan, T. Lassmann, P. Carninci, J. B. Brown, L. Lipovich, J. M. Gonzalez, M. Thomas, C. A. Davis, R. Shiekhattar, T. R. Gingeras, T. J. Hubbard, C. Notredame, J. Harrow, and R. Guigo, "The gencode v7 catalog of human long noncoding rnas: Analysis of their gene structure, evolution, and expression," *Genome Research*, vol. 22, pp. 1775–89, 2012.

[49] T. R. Mercer, D. J. Gerhardt, M. E. Dinger, J. Crawford, C. Trapnell, J. A. Jeddeloh, J. S. Mattick, and J. L. Rinn, "Targeted rna sequencing reveals the deep complexity of the human transcriptome," *Nature Biotechnology*, vol. 30, no. 1, pp. 99–104, 2012.

[50] M. Guttman and J. L. Rinn, "Modular regulatory principles of large non-coding rnas," *Nature*, vol. 482, pp. 339–46, 2012.

[51] R. A. Gupta, N. Shah, K. C. Wang, J. Kim, H. M. Horlings, D. J. Wong, M.-C. Tsai, T. Hung, P. Argani, J. L. Rinn, *et al.*, "Long non-coding rna hotair reprograms chromatin state to promote cancer metastasis," *Nature*, vol. 464, no. 7291, pp. 1071–1076, 2010.

[52] J. T. Kung, D. Colognori, and J. T. Lee, "Long noncoding rnas: Past, present, and future," *Genetics*, vol. 193, pp. 651–69, 2013.

[53] B. Signal, B. S. Gloss, and M. E. Dinger, "Computational approaches for functional prediction and characterisation of long noncoding rnas," *Trends in Genetics*, vol. 32, no. 10, pp. 620–637, 2016.

[54] J. L. Rinn, M. Kertesz, J. K. Wang, S. L. Squazzo, X. Xu, S. A. Brugmann, L. H. Goodnough, J. A. Helms, P. J. Farnham, E. Segal, *et al.*, "Functional demarcation of active and silent chromatin domains in human hox loci by noncoding rnas," *Cell*, vol. 129, no. 7, pp. 1311–1323, 2007.

[55] M.-C. Tsai, O. Manor, Y. Wan, N. Mosammaparast, J. K. Wang, F. Lan, Y. Shi, E. Segal, and H. Y. Chang, "Long noncoding rna as modular scaffold of histone modification complexes," *Science*, vol. 329, no. 5992, pp. 689–693, 2010.

[56] C. Chu, Q. C. Zhang, S. T. Da Rocha, R. A. Flynn, M. Bharadwaj, J. M. Calabrese, T. Magnuson, E. Heard, and H. Y. Chang, "Systematic discovery of xist rna binding proteins," *Cell*, vol. 161, no. 2, pp. 404–416, 2015.

[57] W. Jiang, Y. Liu, R. Liu, K. Zhang, and Y. Zhang, "The lncrna deanr1 facilitates human endoderm differentiation by activating foxa2 expression," *Cell reports*, vol. 11, no. 1, pp. 137–148, 2015.

[58] J.-H. Yoon, K. Abdelmohsen, S. Srikantan, X. Yang, J. L. Martindale, S. De, M. Huarte, M. Zhan, K. G. Becker, and M. Gorospe, "Lincrna-p21 suppresses target mrna translation," *Molecular cell*, vol. 47, no. 4, pp. 648–655, 2012.

[59] T. Hung, Y. Wang, M. F. Lin, A. K. Koegel, Y. Kotake, G. D. Grant, H. M. Horlings, N. Shah, C. Umbricht, P. Wang, *et al.*, "Extensive and coordinated transcription of noncoding rnas within cell-cycle promoters," *Nature genetics*, vol. 43, no. 7, pp. 621–629, 2011.

[60] C. Chu, R. C. Spitale, and H. Y. Chang, "Technologies to probe functions and mechanisms of long noncoding rnas," *Nature structural & molecular biology*, vol. 22, no. 1, pp. 29–35, 2015.

[61] Y. Yang, L. Wen, and H. Zhu, "Unveiling the hidden function of long non-coding rna by identifying its major partner-protein," *Cell & bioscience*, vol. 5, no. 1, p. 1, 2015.

[62] J. L. Rinn and H. Y. Chang, "Genome regulation by long noncoding rnas," *Annual Review of Biochemistry*, vol. 81, pp. 145–66, 2012.

[63] M. Huarte, M. Guttman, D. Feldser, M. Garber, M. J. Koziol, D. Kenzelmann-Broz, A. M. Khalil, O. Zuk, I. Amit, M. Rabani, *et al.*, "A large intergenic noncoding rna induced by p53 mediates global gene repression in the p53 response," *Cell*, vol. 142, no. 3, pp. 409–419, 2010.

[64] G. Housman and I. Ulitsky, "Methods for distinguishing between protein-coding and long noncoding rnas and the elusive biological purpose of translation of long noncoding rnas," *Biochimica et Biophysica Acta*, vol. 1859, pp. 31–40, 2016.

[65] I. Ulitsky, A. Shkumatava, C. H. Jan, H. Sive, and D. P. Bartel, "Conserved function of lincrnas in vertebrate embryonic development despite rapid sequence evolution," *Cell*, vol. 147, no. 7, pp. 1537–1550, 2011.

[66] L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, and Y. Zhao, "Utilizing sequence intrinsic composition to classify protein-coding and long noncoding transcripts," *Nucleic Acids Research*, vol. 41, e166, 2013.

[67] L. Wang, H. J. Park, S. Dasari, S. Wang, J. P. Kocher, and W. Li, "Cpat: Coding-potential assessment tool using an alignment-free logistic regression model," *Nucleic Acids Research*, vol. 41, e74, 2013.

[68] R. Achawanantakun, J. Chen, Y. Sun, and Y. Zhang, "Lncrna-id: Long non-coding rna identification using balanced random forests," *Bioinformatics*, vol. 31, pp. 3897–905, 2015.

[69] L. Kong, Y. Zhang, Z. Q. Ye, X. Q. Liu, S. Q. Zhao, L. Wei, and G. Gao, "Cpc: Assess the protein-coding potential of transcripts using sequence features and support vector machine," *Nucleic Acids Research*, vol. 35, W345–9, 2007.

[70] M. F. Lin, I. Jungreis, and M. Kellis, "Phylocsf: A comparative genomics method to distinguish protein coding and non-coding regions," *Bioinformatics*, vol. 27, pp. i275–82, 2011.

[71] A. Li, J. Zhang, and Z. Zhou, "Plek: A tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme," *BMC Bioinformatics*, vol. 15, p. 1, 2014.

[72] L. Sun, H. Liu, L. Zhang, and J. Meng, "Lncrscan-svm: A tool for predicting long non-coding rnas using support vector machine," *PLOS ONE*, vol. 10, e0139654, 2015.

[73] B. Yan, Z.-H. Wang, and J.-T. Guo, "The research strategies for probing the function of long noncoding rnas," *Genomics*, vol. 99, no. 2, pp. 76–80, 2012.

[74] L. A. Selth, C. Gilbert, and J. Q. Svejstrup, "Rna immunoprecipitation to determine rna-protein associations in vivo," *Cold Spring Harbor Protocols*, vol. 2009, no. 6, pdb–prot5234, 2009.

[75] J. Zhao, B. K. Sun, J. A. Erwin, J.-J. Song, and J. T. Lee, "Polycomb proteins targeted by a short repeat rna to the mouse x chromosome," *Science*, vol. 322, no. 5902, pp. 750–756, 2008.

[76] J. Konig, K. Zarnack, G. Rot, T. Curk, M. Kayikci, B. Zupan, D. J. Turner, N. M. Luscombe, and J. Ule, "Iclip-transcriptome-wide mapping of protein-rna interactions with individual nucleotide resolution," *Journal of visualized experiments: JoVE*, no. 50, 2011.

[77] C. Chu, J. Quinn, and H. Y. Chang, "Chromatin isolation by rna purification (chirp)," *Journal of visualized experiments: JoVE*, no. 61, 2012.

[78] C. Chu, K. Qu, F. L. Zhong, S. E. Artandi, and H. Y. Chang, "Genomic maps of long noncoding rna occupancy reveal principles of rna-chromatin interactions," *Molecular Cell*, vol. 44, no. 4, pp. 667–678, 2011.

[79] L. S. Qi, M. H. Larson, L. A. Gilbert, J. A. Doudna, J. S. Weissman, A. P. Arkin, and W. A. Lim, "Repurposing crispr as an rna-guided platform for sequence-specific control of gene expression," *Cell*, vol. 152, no. 5, pp. 1173–1183, 2013.

[80] S. Konermann, M. D. Brigham, A. E. Trevino, J. Joung, O. O. Abudayyeh, C. Barcena, P. D. Hsu, N. Habib, J. S. Gootenberg, H. Nishimasu, *et al.*, "Genome-scale transcriptional activation by an engineered crispr-cas9 complex," *Nature*, vol. 517, no. 7536, p. 583, 2015.

[81] Q. Jiang, R. Ma, J. Wang, X. Wu, S. Jin, J. Peng, R. Tan, T. Zhang, Y. Li, and Y. Wang, "Lncrna2function: A comprehensive resource for functional investigation of human lncrnas based on rna-seq data," *BMC genomics*, vol. 16, no. 3, S2, 2015.

[82] T. Alam, M. Uludag, M. Essack, A. Salhi, H. Ashoor, J. B. Hanks, C. Kapfer, K. Mineta, T. Gojobori, and V. B. Bajic, "Farna: Knowledgebase of inferred functions of non-coding rna transcripts," *Nucleic acids research*, vol. 45, no. 5, pp. 2838–2848, 2017.

[83] X. Chen, Z.-H. You, G.-Y. Yan, and D.-W. Gong, "Irwrlda: Improved random walk with restart for lncrna-disease association prediction," *Oncotarget*, vol. 7, no. 36, p. 57 919, 2016.

[84] S. He, H. Zhang, H. Liu, and H. Zhu, "Longtarget: A tool to predict lncrna dna-binding motifs and binding sites via hoogsteen base-pairing analysis," *Bioinformatics*, vol. 31, no. 2, pp. 178–186, 2014.

[85] J. Li, W. Ma, P. Zeng, J. Wang, B. Geng, J. Yang, and Q. Cui, "Lnctar: A tool for predicting the rna targets of long noncoding rnas," *Briefings in Bioinformatics*, vol. 16, no. 5, pp. 806–812, 2015.

[86] Q. Lu, S. Ren, M. Lu, Y. Zhang, D. Zhu, X. Zhang, and T. Li, "Computational prediction of associations between long non-coding rnas and proteins," *BMC genomics*, vol. 14, no. 1, p. 1, 2013.

[87] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[88] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in Bioinformatics*, bbw068, 2016.

[89] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 145–154, 2011.

[90] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[91] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[92] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[93] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[94] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, vol. 2, 2010, p. 3.

[95] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[96] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 873–880.

[97] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[98] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[99] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[100] Y.-J. Kang, D.-C. Yang, L. Kong, M. Hou, Y.-Q. Meng, L. Wei, and G. Gao, "Cpc2: A fast and accurate coding potential calculator based on sequence intrinsic features," *Nucleic Acids Research*, 2017.

[101] X. N. Fan and S. W. Zhang, "Lncrna-mfdl: Identification of human long non-coding rnas by fusing multiple features and using deep learning," *Molecular Biosystems*, vol. 11, pp. 892–7, 2015.

[102] L. Hu, Z. Xu, B. Hu, and Z. J. Lu, "Come: A robust coding potential calculation tool for lncrna identification and characterization based on multiple features," *Nucleic Acids Research*, gkw798, 2016.

[103] H. W. Schneider, T. Raiol, M. M. Brigido, M. E.M. T. Walter, and P. F. Stadler, "A support vector machine based method to distinguish long non-coding rnas from protein coding transcripts," *BMC Genomics*, vol. 18, no. 1, p. 804, 2017.

[104] J. Zhao, X. Song, and K. Wang, "Lncscore: Alignment-free identification of long noncoding rna from assembled novel transcripts," *Scientific Reports*, 2016.

[105] V. Wucher, F. Legeai, B. Hedan, G. Rizk, L. Lagoutte, T. Leeb, V. Jagannathan, E. Cadieu, A. David, H. Lohi, *et al.*, "Feelnc: A tool for long non-coding rna annotation and its application to the dog transcriptome," *Nucleic Acids Research*, gkw1306, 2017.

[106] U. K. Muppirala, V. G. Honavar, and D. Dobbs, "Predicting rna-protein interactions using only sequence information," *BMC Bioinformatics*, vol. 12, no. 1, p. 1, 2011.

[107] V Suresh, L. Liu, D. Adjeroh, and X. Zhou, "Rpi-pred: Predicting ncrna-protein interaction using sequence and structural information," *Nucleic Acids Research*, gkv020, 2015.

[108] M. Akbaripour-Elahabad, J. Zahiri, R. Rafeh, M. Eslami, and M. Azari, "Rpicool: A tool for in silico rna–protein interaction detection using random forest," *Journal of Theoretical Biology*, vol. 402, pp. 1–8, 2016.

[109] X. Pan, Y.-X. Fan, J. Yan, and H.-B. Shen, "Ipminer: Hidden ncrna-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction," *BMC genomics*, vol. 17, no. 1, p. 1, 2016.

[110] I. Ulitsky and D. P. Bartel, "Lincrnas: Genomics, evolution, and mechanisms," *Cell*, vol. 154, pp. 26–46, 2013.

[111] M. K. Iyer, Y. S. Niknafs, R. Malik, U. Singhal, A. Sahu, Y. Hosono, T. R. Barrette, J. R. Prensner, J. R. Evans, S. Zhao, A. Poliakov, X. Cao, S. M. Dhanasekaran, Y. M. Wu, D. R. Robinson, D. G. Beer, F. Y. Feng, H. K. Iyer, and A. M. Chinnaiyan, "The landscape of long noncoding rnas in the human transcriptome," *Nature Genetics*, vol. 47, pp. 199–208, 2015.

[112] R. D. Finn, J. Clements, and S. R. Eddy, "Hmmer web server: Interactive sequence similarity searching," *Nucleic Acids Research*, gkr367, 2011.

[113] R. Lorenz, S. H. Bernhart, C. H. Zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker, "Viennarna package 2.0," *Algorithms for Molecular Biology*, vol. 6, no. 1, p. 1, 2011.

[114] K. D. Pruitt, T. Tatusova, G. R. Brown, and D. R. Maglott, "Ncbi reference sequences (refseq): Current status, new features and genome annotation policy," *Nucleic Acids Research*, vol. 40, no. D1, pp. D130–D135, 2012.

[115] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, *et al.*, "Gencode: The reference human genome annotation for the encode project," *Genome Research*, vol. 22, no. 9, pp. 1760–1774, 2012.

[116] W. Li and A. Godzik, "Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[117] H. Zhu, G.-Q. Hu, Y.-F. Yang, J. Wang, and Z.-S. She, "Med: A new non-supervised gene prediction algorithm for bacterial and archaeal genomes," *BMC Bioinformatics*, vol. 8, no. 1, p. 97, 2007.

[118] Y. Liu, J. Guo, G. Hu, and H. Zhu, "Gene prediction in metagenomic fragments based on the svm algorithm," *BMC Bioinformatics*, vol. 14, no. Suppl 5, S12, 2013.

[119] J. W. Fickett and C.-S. Tung, "Assessment of protein coding measures," *Nucleic Acids Research*, vol. 20, no. 24, pp. 6441–6450, 1992.

[120] H. Liu, J. Yin, M. Xiao, C. Gao, A. S. Mason, Z. Zhao, Y. Liu, J. Li, and D. Fu, "Characterization and evolution of 5 and 3 untranslated regions in eukaryotes," *Gene*, vol. 507, no. 2, pp. 106–111, 2012.

[121] J. W. Fickett, "Recognition of protein coding regions in dna sequences," *Nucleic Acids Research*, vol. 10, pp. 5303–5318, 1982.

[122] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, *et al.*, "The pfam protein families database: Towards a more sustainable future," *Nucleic Acids Research*, vol. 44, no. D1, pp. D279–D285, 2016.

[123] J. Bentley, "Programming pearls: Algorithm design techniques," *Communications of the ACM*, vol. 27, no. 9, pp. 865–873, Sep. 1984.

[124] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[125] A. Rhoads and K. F. Au, "Pacbio sequencing and its applications," *Genomics, proteomics & bioinformatics*, vol. 13, no. 5, pp. 278–289, 2015.

[126] D. Branton, D. W. Deamer, A. Marziali, H. Bayley, S. A. Benner, T. Butler, M. Di Ventra, S. Garaj, A. Hibbs, X. Huang, *et al.*, "The potential and challenges of nanopore sequencing," *Nature biotechnology*, vol. 26, no. 10, pp. 1146–1153, 2008.

[127] V. Murigneux, J. Saulière, H. R. Crollius, and H. Le Hir, "Transcriptome-wide identification of rna binding sites by clip-seq," *Methods*, vol. 63, no. 1, pp. 32–40, 2013.

[128] J. König, K. Zarnack, N. M. Luscombe, and J. Ule, "Protein–rna interactions: New genomic technologies and perspectives," *Nature Reviews. Genetics*, vol. 13, no. 2, pp. 77–83, 2012.

[129] M. Bellucci, F. Agostini, M. Masin, and G. G. Tartaglia, "Predicting protein associations with long noncoding rnas," *Nature Methods*, vol. 8, no. 6, pp. 444–445, 2011.

[130] D. Cirillo, M. Blanco, A. Armaos, A. Buness, P. Avner, M. Guttman, A. Cerase, and G. G. Tartaglia, "Quantitative predictions of protein interactions with long noncoding rnas," *Nature Methods*, vol. 14, no. 1, pp. 5–6, 2017.

[131] A. Li, M. Ge, Y. Zhang, C. Peng, and M. Wang, "Predicting long noncoding rna and protein interactions using heterogeneous network model," *Biomed Research International*, vol. 2015, 2015.

[132] J. Yang, A. Li, M. Ge, and M. Wang, "Relevance search for predicting lncrna–protein interactions based on heterogeneous network," *Neurocomputing*, 2016.

[133] M. Ge, A. Li, and M. Wang, "A bipartite network-based method for prediction of long non-coding rna–protein interactions," *Genomics, proteomics & bioinformatics*, vol. 14, no. 1, pp. 62–71, 2016.

[134] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa, "Kegg for representation and analysis of molecular networks involving diseases and drugs," *Nucleic Acids Research*, vol. 38, no. suppl 1, pp. D355–D360, 2010.

[135] D. Croft, A. F. Mundo, R. Haw, M. Milacic, J. Weiser, G. Wu, M. Caudy, P. Garapati, M. Gillespie, M. R. Kamdar, *et al.*, "The reactome pathway knowledgebase," *Nucleic Acids Research*, vol. 42, no. D1, pp. D472–D477, 2014.

[136] J. Yuan, W. Wu, C. Xie, G. Zhao, Y. Zhao, and R. Chen, "Npinter v2.0: An updated database of ncrna interactions," *Nucleic Acids Research*, vol. 42, no. D1, pp. D104–D108, 2014.

[137] D. Bu, K. Yu, S. Sun, C. Xie, G. Skogerbø, R. Miao, H. Xiao, Q. Liao, H. Luo, G. Zhao, *et al.*, "Noncode v3.0: Integrative annotation of long noncoding rnas," *Nucleic Acids Research*, gkr1175, 2011.

[138] U. Consortium *et al.*, "Reorganizing the protein space at the universal protein resource (uniprot)," *Nucleic Acids Research*, gkr981, 2011.

[139] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4337–4341, 2007.

[140] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[141] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[142] B. Zhang, D. T. Yehdego, K. L. Johnson, M.-Y. Leung, and M. Taufer, "Enhancement of accuracy and efficiency for rna secondary structure prediction by sequence segmentation and mapreduce," *BMC Structural Biology*, vol. 13, no. 1, S3, 2013.

[143] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, IEEE, 2012, pp. 2133–2136.

[144] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[145] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, *et al.*, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005.

[146] D. Frishman and P. Argos, "Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence," *Protein Engineering*, vol. 9, no. 2, pp. 133–142, 1996.

[147] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the royal statistical society. Series B (Methodological)*, pp. 289–300, 1995.

[148] X. Tang, J. Wang, B. Liu, M. Li, G. Chen, and Y. Pan, "A comparison of the functional modules identified from time course and static ppi network data," *BMC Bioinformatics*, vol. 12, no. 1, p. 339, 2011.

[149] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Research*, vol. 30, no. 7, pp. 1575–1584, 2002.

[150] G. Alanis-Lobato, M. A. Andrade-Navarro, and M. H. Schaefer, "Hippie v2. 0: Enhancing meaningfulness and reliability of protein–protein interaction networks," *Nucleic Acids Research*, gkw985, 2016.

[151] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal, Complex Systems*, vol. 1695, no. 5, pp. 1–9, 2006.

[152] M. Hajjari and A. Salavaty, "Hotair: An oncogenic long non-coding rna in different cancers," *Cancer biology & medicine*, vol. 12, no. 1, p. 1, 2015.

[153] M. Lee, H. J. Kim, S. W. Kim, S.-A. Park, K.-H. Chun, N. H. Cho, Y. S. Song, and Y. T. Kim, "The long non-coding rna hotair increases tumour growth and invasion in cervical cancer by targeting the notch pathway," *Oncotarget*, vol. 7, no. 28, p. 44 558, 2016.

[154] X.-S. Ge, H.-J. Ma, X.-H. Zheng, H.-L. Ruan, X.-Y. Liao, W.-Q. Xue, Y.-B. Chen, Y. Zhang, and W.-H. Jia, "Hotair, a prognostic factor in esophageal squamous cell carcinoma, inhibits wif-1 expression and activates wnt pathway," *Cancer Sci.*, vol. 104, no. 12, pp. 1675–1682, 2013.

[155] H. Zhang and X. Meng, "Association of hotair expression with pi3k/akt pathway activation in adenocarcinoma of esophagogastric junction," *Open Medicine*, vol. 11, p. 36, 2016.

[156] E. Pasmant, I. Laurendeau, D. Héron, M. Vidaud, D. Vidaud, and I. Bieche, "Characterization of a germ-line deletion, including the entire ink4/arf locus, in a melanoma-neural system tumor family: Identification of anril, an antisense noncoding rna whose expression coclusters with arf," *Cancer Research*, vol. 67, no. 8, pp. 3963–3969, 2007.

[157] M. Iranpour, M. Soudyab, L. Geranpayeh, R. Mirfakhraie, E. Azargashb, A. Movafagh, and S. Ghafouri-Fard, "Expression analysis of four long noncoding rnas in breast cancer," *Tumor Biology*, vol. 37, no. 3, pp. 2933–2940, 2016.

[158] E. Pasmant, A. Sabbagh, M. Vidaud, and I. Bièche, "Anril, a long, noncoding rna, is an unexpected major hotspot in gwas," *The FASEB Journal*, vol. 25, no. 2, pp. 444–448, 2011.

[159] L. Folkersen, T. Kyriakou, A. Goel, J. Peden, A. Mälarstig, G. Paulsson-Berne, A. Hamsten, A. Franco-Cereceda, A. Gabrielsen, P. Eriksson, *et al.*, "Relationship between cad risk genotype in the chromosome 9p21 locus and gene expression. identification of eight new anril splice variants," *PLOS ONE*, vol. 4, no. 11, e7677, 2009.

[160] H. Nakaoka, A. Gurumurthy, T. Hayano, S. Ahmadloo, W. H. Omer, K. Yoshihara, A. Yamamoto, K. Kurose, T. Enomoto, S. Akira, *et al.*, "Allelic imbalance in regulation of anril through chromatin interaction at 9p21 endometriosis risk locus," *PLoS Genetics*, vol. 12, no. 4, e1005893, 2016.

[161] D. Khaitan, M. E. Dinger, J. Mazar, J. Crawford, M. A. Smith, J. S. Mattick, and R. J. Perera, "The melanoma-upregulated long noncoding rna spry4-it1 modulates apoptosis and invasion," *Cancer Research*, vol. 71, no. 11, pp. 3852–3862, 2011.

[162] Y. Shi, J. Li, Y. Liu, J. Ding, Y. Fan, Y. Tian, L. Wang, Y. Lian, K. Wang, and Y. Shu, "The long noncoding rna spry4-it1 increases the proliferation of human breast cancer cells by upregulating znf703 expression," *Molecular Cancer*, vol. 14, no. 1, p. 51, 2015.

[163] M. Sand, F. G. Bechara, D. Sand, T. Gambichler, S. A. Hahn, M. Bromba, E. Stockfleth, and S. Hessam, "Long-noncoding rnas in basal cell carcinoma," *Tumor Biology*, vol. 37, no. 8, pp. 10 595–10 608, 2016.

[164] Q. Zuo, S. Huang, Y. Zou, Y. Xu, Z. Jiang, S. Zou, H. Xu, and L. Sun, "The lnc rna spry4-it1 modulates trophoblast cell invasion and migration by affecting the epithelial-mesenchymal transition," *Scientific Reports*, vol. 6, p. 37 183, Nov. 2016.

[165] N. Lin, K.-Y. Chang, Z. Li, K. Gates, Z. A. Rana, J. Dang, D. Zhang, T. Han, C.-S. Yang, T. J. Cunningham, *et al.*, "An evolutionarily conserved long noncoding rna tuna controls pluripotency and neural lineage commitment," *Molecular Cell*, vol. 53, no. 6, pp. 1005–1019, 2014.

[166] I. Morn,. Akerman, M. vandeBunt, R. Xie, M. Benazra, T. Nammo, L. Arnes, N. Naki, J. Garca-Hurtado, S. Rodrguez-Segu, L. Pasquali, C. Sauty-Colace, A. Beucher, R. Scharfmann, J. vanArensbergen, P. Johnson, A. Berry, C. Lee, T. Harkins, V. Gmyr, F. Pattou, J. Kerr-Conte, L. Piemonti, T. Berney, N. Hanley, A. Gloyn, L. Sussel, L. Langman, K. Brayman, M. Sander, M. McCarthy, P. Ravassard, and J. Ferrer, "Human beta cell transcriptome analysis uncovers lncrnas that are tissue-specific, dynamically regulated, and abnormally expressed in type 2 diabetes," *Cell Metabolism*, vol. 16, no. 4, pp. 435–448, 2012.

[167] H. Li, L. Zhu, L. Xu, K. Qin, C. Liu, Y. Yu, D. Su, K. Wu, and Y. Sheng, "Long noncoding rna linc00617 exhibits oncogenic activity in breast cancer," *Molecular Carcinogenesis*, 2015.

[168] Q. Jiang, R. Ma, J. Wang, X. Wu, S. Jin, J. Peng, R. Tan, T. Zhang, Y. Li, and Y. Wang, "Lncrna2function: A comprehensive resource for functional investigation of human lncrnas based on rna-seq data," *BMC Genomics*, vol. 16 Suppl 3, S2, 2015.

[169] T. Alam, M. Uludag, M. Essack, A. Salhi, H. Ashoor, J. B. Hanks, C. Kapfer, K. Mineta, T. Gojobori, and V. B. Bajic, "Farna: Knowledgebase of inferred functions of non-coding rna transcripts," *Nucleic Acids Research*, gkw973, 2016.

[170] Z. Cheng, S. Zhou, and J. Guan, "Computationally predicting protein-rna interactions using only positive and unlabeled examples," *Journal of Bioinformatics and Computational Biology*, vol. 13, no. 03, p. 1 541 005, 2015.

[171] H.-P. Sinn and H. Kreipe, "A brief overview of the who classification of breast tumors," *Breast Care*, vol. 8, no. 2, pp. 149–154, 2013.

[172] M. C. Abba, T. Gong, Y. Lu, J. Lee, Y. Zhong, E. Lacunza, M. Butti, Y. Takata, S. Gaddis, J. Shen, *et al.*, "A molecular portrait of high-grade ductal carcinoma in situ," *Cancer research*, vol. 75, no. 18, pp. 3980–3990, 2015.

[173] K. E. Varley, J. Gertz, B. S. Roberts, N. S. Davis, K. M. Bowling, M. K. Kirby, A. S. Nesmith, P. G. Oliver, W. E. Grizzle, A. Forero, *et al.*, "Recurrent read-through fusion transcripts in breast cancer," *Breast cancer research and treatment*, vol. 146, no. 2, pp. 287–297, 2014.

[174] Y. Tuo, X. Li, and J Luo, "Long noncoding rna uca1 modulates breast cancer cell growth and apoptosis through decreasing tumor suppressive mir-143," *European Review for Medical and Pharmacological Sciences*, vol. 19, no. 18, pp. 3403–11, 2015.

[175] F. Rapaport, R. Khanin, Y. Liang, M. Pirun, A. Krek, P. Zumbo, C. E. Mason, N. D. Socci, and D. Betel, "Comprehensive evaluation of differential gene expression analysis methods for rna-seq data," *Genome Biology*, vol. 14, no. 9, p. 3158, 2013.

[176] G. R. Grant, M. H. Farkas, A. D. Pizarro, N. F. Lahens, J. Schug, B. P. Brunk, C. J. Stoeckert, J. B. Hogenesch, and E. A. Pierce, "Comparative analysis of rna-seq alignment algorithms and the rna-seq unified mapper (rum)," *Bioinformatics*, vol. 27, no. 18, pp. 2518–2528, 2011.

[177] P.-Y. Wu, J. H. Phan, and M. D. Wang, "An approach for assessing rna-seq quantification algorithms in replication studies," in *Genomic Signal Processing and Statistics (GENSIPS), 2013 IEEE International Workshop on*, IEEE, 2013, pp. 15–18.

[178] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[179] T. D. Wu and S. Nacu, "Fast and snp-tolerant detection of complex variants and splicing in short reads," *Bioinformatics*, vol. 26, no. 7, pp. 873–881, 2010.

[180] *Novoalign*, `http://www.novocraft.com`.

[181] S.-I. Consortium *et al.*, "A comprehensive assessment of rna-seq accuracy, reproducibility and information content by the sequencing quality control consortium," *Nature Biotechnology*, vol. 32, no. 9, pp. 903–914, 2014.

[182] D. Thierry-Mieg and J. Thierry-Mieg, "Aceview: A comprehensive cdna-supported gene and transcripts annotation," *Genome Biology*, vol. 7, no. 1, S12, 2006.

[183] J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad, "Rna-seq: An assessment of technical reproducibility and comparison with gene expression arrays," *Genome Research*, vol. 18, no. 9, pp. 1509–1517, 2008.

[184] Z. Peng, Y. Cheng, B. C.-M. Tan, L. Kang, Z. Tian, Y. Zhu, W. Zhang, Y. Liang, X. Hu, X. Tan, *et al.*, "Comprehensive analysis of rna-seq data reveals extensive rna editing in a human transcriptome," *Nature Biotechnology*, vol. 30, no. 3, pp. 253–260, 2012.

[185] A. Oshlack, M. D. Robinson, and M. D. Young, "From rna-seq reads to differential expression results," *Genome Biology*, vol. 11, no. 12, p. 220, 2010.

[186] O. D. Iancu, S. Kawane, D. Bottomly, R. Searles, R. Hitzemann, and S. McWeeney, "Utilizing rna-seq data for de novo coexpression network inference," *Bioinformatics*, vol. 28, no. 12, pp. 1592–1597, 2012.

[187] N. A. Fonseca, J. Marioni, and A. Brazma, "Rna-seq gene profiling-a systematic empirical comparison," *PLOS ONE*, vol. 9, no. 9, e107026, 2014.

[188] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," *Briefings in Bioinformatics*, vol. 11, no. 5, pp. 473–483, 2010.

[189] R. Chandramohan, P.-Y. Wu, J. H. Phan, and M. D. Wang, "Systematic assessment of rna-seq quantification tools using simulated sequence data," in *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, ACM, 2013, p. 623.

[190] J. A. Robles, S. E. Qureshi, S. J. Stephen, S. R. Wilson, C. J. Burden, and J. M. Taylor, "Efficient experimental design and analysis strategies for the detection of dif-

ferential expression using rna-sequencing," *BMC genomics*, vol. 13, no. 1, p. 484, 2012.

[191] P. G. Engström, T. Steijger, B. Sipos, G. R. Grant, A. Kahles, G. Rätsch, N. Goldman, T. J. Hubbard, J. Harrow, R. Guigó, *et al.*, "Systematic evaluation of spliced alignment programs for rna-seq data," *Nature Methods*, vol. 10, no. 12, pp. 1185–1191, 2013.

[192] B. Sipos, G. Slodkowicz, T. Massingham, and N. Goldman, "Realistic simulations reveal extensive sample-specificity of rna-seq biases," *arXiv preprint arXiv:1308.3172*, 2013.

[193] *Simngs–software for simulating next-generation sequencing data*, `http://www.ebi.ac.uk/goldman-srv/simNGS/`.

[194] X. Zheng and E. N. Moriyama, "Comparative studies of differential gene calling using rna-seq data," *BMC Bioinformatics*, vol. 14, no. 13, S7, 2013.

[195] S. C. Munger, N. Raghupathy, K. Choi, A. K. Simons, D. M. Gatti, D. A. Hinerfeld, K. L. Svenson, M. P. Keller, A. D. Attie, M. A. Hibbs, *et al.*, "Rna-seq alignment to individualized genomes improves transcript abundance estimates in multiparent populations," *Genetics*, vol. 198, no. 1, pp. 59–73, 2014.

[196] J. Wu, O. Anczuków, A. R. Krainer, M. Q. Zhang, and C. Zhang, "Olego: Fast and sensitive mapping of spliced mrna-seq reads using small seeds," *Nucleic Acids Research*, vol. 41, no. 10, pp. 5149–5163, 2013.

[197] Y. Liao, G. K. Smyth, and W. Shi, "The subread aligner: Fast, accurate and scalable read mapping by seed-and-vote," *Nucleic Acids Research*, vol. 41, no. 10, e108–e108, 2013.

[198] S. Huang, J. Zhang, R. Li, W. Zhang, Z. He, T.-W. Lam, Z. Peng, and S.-M. Yiu, "Soapsplice: Genome-wide ab initio detection of splice junctions from rna-seq data," *Frontiers in Genomic Assay Technology*, 2011.

VITA

Cheng Yang was born in Jiangxi Province, China in 1990. He received his Bachelor's degree from College of Engineering, Peking University in 2011. Afterward, he participated in the PKU/GT/Emory joint Ph.D. program and majored in Biomedical Engineering. From January 2014 to July 2015, he studied at Georgia Institute of Technology. During Ph.D., he developed tools and algorithms for bioinformatics analysis. His research interests lie in bioinformatics and health informatics.